

**COMPARACIÓN DE METAHEURÍSTICAS APLICADAS AL PROBLEMA DE
CORTE BIDIMENSIONAL TIPO GUILLOTINA EN LÁMINAS DE ACERO**



SAIDE AURA RACERO MORELO

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE INGENIERÍAS

DEPARTAMENTO DE INGENIERIA INDUSTRIAL

MONTERÍA

2017

**COMPARACIÓN DE METAHEURÍSTICAS APLICADAS AL PROBLEMA DE
CORTE BIDIMENSIONAL TIPO GUILLOTINA EN LÁMINAS DE ACERO**

SAIDE AURA RACERO MORELO

Trabajo de grado presentado en la modalidad de Trabajo de Investigación y/o Extensión,
como parte de los requisitos para optar al Título de Ingeniero Industrial.

Director:

M.Sc. Jorge Mario López Pereira

UNIVERSIDAD DE CÓRDOBA

FACULTAD DE INGENIERÍAS

DEPARTAMENTO DE INGENIERIA INDUSTRIAL

MONTERÍA

2017

Nota de aceptación

M.Sc. Jorge Mario López Pereira

Director

JURADO

JURADO

A Dios

A mis padres Alvaro y Saide

Agradecimiento especial a:

Dios por llenarme de sabiduría, ser mi guía, nunca abandonarme en este camino y enseñarme que con disciplina todo lo que se propone es posible lograrlo.

Profesor Jorge Mario López Pereira, director de esta investigación. Por transmitirme parte de sus conocimientos y parte de su valioso tiempo para asesorarme.

Agradecimientos:

A mis padres Alvaro Racero Raveles y Saide Morelo Cardoza, por darme buena educación, creer en mí, por enseñarme buenos valores y querer siempre lo mejor para mí. Dios no se equivocó en dármeles como Padres, para mi siempre serán los mejores.

A mis hermanas Johana Racero Morelo y Yaniet Racero Morelo, por su apoyo y amor que me brindan cada vez que las necesito.

A mi hermano Cristian Racero, que desde el cielo sé que me transmite mucho amor.

A mis abuelitas que junto a la derecha de Dios me llenan de bendiciones y mucha fuerza para alcanzar mis sueños, y demás familiares por su gran amor.

A mis amigos, aquellos con los que compartí conocimientos durante la carrera Y aquellos que me brindan su sincera amistad.

CONTENIDO

| | |
|---|----|
| CAPITULO 1: GENERALIDADES | 13 |
| 1.1. INTRODUCCIÓN..... | 13 |
| 1.2. PLANTEAMIENTO DEL PROBLEMA..... | 14 |
| 1.2.1. DESCRIPCIÓN DEL PROBLEMA | 14 |
| 1.2.2. SÍNTOMAS DEL PROBLEMA..... | 15 |
| 1.2.3. CAUSAS DEL PROBLEMA | 16 |
| 1.2.4. FORMULACIÓN DEL PROBLEMA..... | 16 |
| 1.2.5. SISTEMATIZACIÓN DE LA INVESTIGACIÓN | 17 |
| 1.3. OBJETIVOS..... | 19 |
| 1.3.1. OBEJTIVO GENERAL..... | 19 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 19 |
| 1.4. JUSTIFICACIÓN..... | 20 |
| CAPITULO 2: MARCO DE REFERENCIA TEÓRICO GENERAL | 22 |
| 2.1. MARCO TEÓRICO | 22 |
| 2.1.1. CORTE Y EMPAQUETAMIENTO (C&P)..... | 22 |
| 2.1.2. DEFINICIÓN DEL PROBLEMA DE CORTE DE DOS DIMENSIONES | 25 |
| 2.1.3. EL PROBLEMA DE CORTE POR GUILLOTINA | 27 |

| | | |
|---|---|----|
| 2.1.4. | CLASIFICACIÓN DEL PROBLEMA DE CORTE | 29 |
| 2.1.5. | PROBLEMAS DE CORTES BIDIMENSIONALES NO LIMITADOS (U2DCP) | 33 |
| 2.1.6. | ALGORITMIA | 34 |
| 2.1.7. | COMPLEJIDAD ALGORITMICA: PROBLEMAS NP-DURO | 35 |
| 2.1.8. | MÉTODOS DE SOLUCIÓN DEL PROBLEMA BIDIMENSIONAL TIPO GUILLOTINA | 36 |
| 2.2. | ESTADO DEL ARTE | 39 |
| 2.2.1. | HEURÍSTICAS APLICADAS AL PROBLEMA DE CORTE BIDIMENSIONAL TIPO GUILLOTINA | 45 |
| 2.2.2. | METAHEURISTICA APLICADAS AL PROBLEMA BIDIMENSIONAL TIPO GUILLOTINA | 46 |
| CAPITULO 3: METODOLOGÍA | | 49 |
| 3.1. | REPRESENTACIÓN DE LAS SOLUCIONES (ENCODING)..... | 50 |
| 3.2. | DESCRIPCIÓN DE LOS ALGORITMOS | 51 |
| 3.2.1. | Algoritmo Genético (Genetic algorithm: GA)..... | 51 |
| 3.2.2. | Búsqueda Tabú (Tabu Search-TS)..... | 60 |
| 3.2.3. | Optimización del Intercambio Térmico (TEO) | 64 |
| CAPITULO 4: ANÁLISIS DE RESULTADOS Y CONCLUSIONES | | 71 |
| 4.1. | ANÁLISIS DE RESULTADOS | 71 |

| | |
|--|----|
| 4.2. ANÁLISIS DE COMPARACIÓN POR NÚMERO DE PIEZAS EN LAS INSTANCIAS | 75 |
| 4.3. CONCLUSIONES Y TRABAJOS FUTUROS | 83 |
| CAPITULO 5: BIBLIOGRAFÍA..... | 85 |

LISTA DE ILUSTRACIONES

| | |
|---|-----------|
| <i>Ilustración 1: Estructura general del problema de corte bidimensional.....</i> | <i>24</i> |
| <i>Ilustración 2: Patrones de corte en 2 dimensiones.....</i> | <i>25</i> |
| <i>Ilustración 3: Modelo de corte 2D por guillotina.....</i> | <i>28</i> |
| <i>Ilustración 4: Corte de guillotina de dos etapas.....</i> | <i>28</i> |
| <i>Ilustración 5: Ejemplo de encoding de una solución.....</i> | <i>51</i> |
| <i>Ilustración 6: Diagrama del Algoritmo Genético.....</i> | <i>53</i> |
| <i>Ilustración 7: Cruzamiento para la parte binaria del enconding.....</i> | <i>55</i> |
| <i>Ilustración 8: Cruzamiento SJ2OX para un encoding de 8 instalaciones.....</i> | <i>57</i> |
| <i>Ilustración 9: Operador Shift para un encoding de 9 piezas.....</i> | <i>59</i> |
| <i>Ilustración 10: Diagrama del Algoritmo de Búsqueda Tabú.....</i> | <i>62</i> |
| <i>Ilustración 11: Operador Swap para un encoding de 9 piezas</i> | <i>63</i> |
| <i>Ilustración 12: Diagrama de Optimización del intercambio térmico.....</i> | <i>66</i> |

LISTA DE TABLAS

| | |
|--|-----------|
| <i>Tabla 1: Principales operadores de cruzamiento de encodings permutados</i> | <i>56</i> |
| <i>Tabla 2: Valores usados de los parámetros del Algoritmo Genético</i> | <i>60</i> |
| <i>Tabla 3: Valores usados de los parámetros del Algoritmo TEO</i> | <i>70</i> |
| <i>Tabla 4: Comparación de Algoritmos por Instancias proporcionadas por la literatura</i> | <i>72</i> |
| <i>Tabla 5: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS H y HZ1</i> | <i>75</i> |
| <i>Tabla 6: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS M1, M2, M3, M4, M5, U1, U2.....</i> | <i>76</i> |
| <i>Tabla 7: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS UU1, UU3, UU11.....</i> | <i>77</i> |
| <i>Tabla 8: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP12, ATP18, ATP19, B, UU2, UU4, UU6.....</i> | <i>78</i> |
| <i>Tabla 9: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP14, ATP15, U4.....</i> | <i>79</i> |
| <i>Tabla 10: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP10, ATP 11, ATP13, ATP16, ATP17, UU5, UU7, UU8, UU9, UU10</i> | <i>80</i> |

RESUMEN

Muchas industrias del sector de metal, vidrio, madera, papel, entre otras. Que se dedican al corte en dos dimensiones y utilizan una herramienta llamada guillotina, los cuales son estimados como cortes rectangulares tienen un problema común que tiene que ver con el gran número de posibles patrones de cortes que pueden ocurrir al momento de hacer el despiece de material, es decir cuando se desee cortar piezas solicitadas por los clientes en objetos grandes (láminas de acero), y cuyo principal objetivo es minimizar desperdicio de su materia prima. En esta investigación se trata con el caso de que el corte es tipo guillotina, y donde las piezas pueden tener rotación de 90 grados al momento de ser cortadas de la lámina de acero. En esta investigación se presenta una comparación de metaheurísticas que resuelven este problema. Las metaheurísticas utilizadas fueron: Genético, Búsqueda tabú y Optimización del Intercambio Térmico, que mediante un análisis de resultados es escogida la metaheurística que revele mejor resultados (minimización de desperdicio).

Palabras claves: genético, Búsqueda Tabú, Optimización de Intercambio Térmico, corte y empaquetado 2D, rotación, guillotina

Abstract

Many industries in the sector of metal, glass, wood, paper, among others. That they dedicate to the cut in two dimensions and that they use a tool called guillotine, that have problems like rectangular cuts have a common problem that has to do with the great number of possible patterns of cuts that can happen at the moment of making the cutting of material , that is to say when it is sought to cut pieces requested by customers in large objects (steel sheets), and whose main objective is to minimize waste of their raw material. This investigation deals with the case that the cut is guillotine type, and where the pieces can have rotation of 90 degrees at the time of being cut from the steel sheet. This research presents a comparison of metaheuristics that solve this problem. The metaheuristics were: Genetic, Taboo Search and Optimization of Thermal Exchange, which by means of an analysis of results is chosen the metaheuristic that reveals better results (minimization of waste).

Key words: genetic, Taboo Search, Thermal Exchange Optimization, 2D cutting and packaging, rotation, guillotine

CAPITULO 1: GENERALIDADES

1.1.INTRODUCCIÓN

El problema de corte bidimensional (2DCP, Two-dimensional Cutting Problem) trata de cortar un conjunto de piezas (rectángulos) con longitudes conocidas, a partir de una o varias láminas (rectángulos mayores), buscando en lo posible que se utilice el menor número de láminas y así reducir la cantidad de residuos (desperdicios de láminas de acero). Este tipo de problema corresponde a la categoría de los llamados problemas de corte y empaquetamiento (C&P), los cuales son calificados problemas de optimización combinatoria NP-duros.(Furini & Malaguti, 2013) La dificultad inherente de este problema de corte en dos dimensiones, es el gran número de posibles patrones de corte que pueden ocurrir. Los cortes en dos dimensiones se clasifican según el tipo de herramienta a utilizar como son el corte tipo guillotina y corte no guillotina, estimados como cortes rectangulares aplicado en el sector de metal, vidrio, madera, papel. Los cortes tipo guillotina son rectos y de extremo a extremo de la lámina y siempre paralela a uno de los lados de la misma.

En este trabajo este problema tiene las siguientes características:

- El corte es tipo guillotina es decir que el corte va de un extremo hacia el otro extremo de la lámina, ósea que se hace un solo corte, entonces la línea imaginaria de corte deberá no cruzar por la mitad a alguna pieza,
- Se permite el giro de las láminas exclusivamente en 90 grados, es decir de una orientación vertical a una horizontal, o viceversa, y

- Las piezas son estrictamente rectangulares, es decir que no se aceptan piezas circulares, triangulares o irregulares

En esta investigación se examinan los diferentes algoritmos utilizados para solucionar el problema de corte en dos dimensiones tipo guillotina, en el cual son escogidos tres algoritmos para posteriormente ser comparados y así escoger el que mejor arroje soluciones óptimas. Estos algoritmos son programados en el software Matlab igualmente son resueltos por este software para diferentes instancias del problema encontradas en la literatura, realizando un análisis de los resultados obtenidos.

1.2.PLANTEAMIENTO DEL PROBLEMA

1.2.1. DESCRIPCIÓN DEL PROBLEMA

En un mundo globalizado como el de hoy, las empresas andan tras la búsqueda de una estrategia competitiva que les ayude hacer la diferencia. La reducción de los costos, disminución de los desperdicios, bajas en los precios de ventas y aumento de la calidad del producto son algunos de los objetivos que persiguen las empresas para obtener un lugar en el mercado, el cual les permita generar un aumento en la rentabilidad y en las utilidades de la empresa. En Sectores como el papel, cuero, vidrio, metales, madera, entre otros, se realizan procedimientos de cortes en el que muchas veces no se establecen los métodos adecuados para obtener un patrón de corte que permita realizar el trabajo, generando un mínimo desperdicio de materia prima y el cual se pueda obtener en un lapso corto de tiempo. Es por esto, que los problemas de corte aplicados a este tipo de industria se han venido

estudiando con la finalidad de aportar herramientas que ayuden a la optimización del proceso de corte cuando se trabaja especialmente en dos dimensiones.

Hoy en día en el mundo se le está dando gran importancia a la creación y al desarrollo de nuevos algoritmos optimizadores que ayuden a solucionar y resolver problemas que se han considerado de gran complejidad o magnitud, ya sea debido a que estos problemas posean gran cantidad de variables o por las innumerables restricciones que se encuentren en los mismos, de una u otra forma la necesidad de estas herramientas es apremiante para encontrar soluciones optimas en diferentes casos de optimización y a los distintos problemas que en la actualidad lo requieran.

Este tipo de algoritmos llamados metaheurísticas son verdaderamente necesarios en nuestra realidad actual, ya que gracias a ellos se pueden encontrar soluciones a muchos problemas que se presentan diariamente y que son considerados imposibles de resolver en cortos periodos de tiempo. Sin embargo, se están encontrando muchos inconvenientes, limitaciones o restricciones al momento de hacer uso de estos, ya que sus aplicaciones en casos reales están siendo muy afectadas debido a que estos algoritmos todavía requieren de un mayor rendimiento y eficiencia en la búsqueda de sus resultados.

1.2.2. SÍNTOMAS DEL PROBLEMA

La falta de conocimiento para adaptar algoritmos que solucionen el problema de corte bidimensional tipo guillotina teniendo en cuenta que la elaboración a mano de los cortes de las piezas solicitadas por los clientes genera en las empresas baja productividad y poca eficiencia a través del alto costo de producción, desperdicios

de láminas de acero, pérdida de tiempo por parte del jefe de producción y demora en la entrega de pedidos a los clientes.

Organizaciones que no generen productividad y que sean poco eficientes dejan de ser competitivas y poco rentables, dejando así que los clientes se vayan a otras organizaciones del mismo sector convirtiendo a estas más fuertes.

1.2.3. CAUSAS DEL PROBLEMA

Los factores que son las causas de este tipo de problemas son:

- Mucho tiempo al elaborar los patrones de corte,
- No se tiene eficiencia al momento de realizar el despiece de material
- No se consta con una herramienta que ayude a optimizar este problema
- Mucho costo de producción

1.2.4. FORMULACIÓN DEL PROBLEMA

Esta investigación debe ofrecer una propuesta metodológica para una correcta exploración a la hora de estudiar el problema de corte bidimensional tipo guillotina. Se emplearán algoritmos que solucionen este tipo de problema, estudiando y comparando los resultados obtenidos y así se podrá responder a la siguiente pregunta:

¿Qué algoritmo fundamentado en los cortes bidimensionales tipo guillotina para láminas de acero tubo un mayor desempeño porque generó mejores resultados con calidad y a un bajo costo computacional?

1.2.5. SISTEMATIZACIÓN DE LA INVESTIGACIÓN

Durante el desarrollo de esta investigación se debe seguir ciertos pasos para lograr los objetivos que se tienen planeados. Es por eso que se han planteado una serie de interrogantes, que a la medida que vayan siendo resueltos se llegara a los resultados que se esperan.

1.2.5.1. Estado del arte y criterios de evaluación

En aras de estudiar el problema en cuestión de la mejor manera posible, se debe conocer y entender las directrices que ha tenido durante los últimos años, es ahí donde surge el siguiente interrogante:

¿Cuál es el estado del arte del problema de corte bidimensional tipo guillotina?

Ahora bien, al conocerse que el problema es de alta complejidad computacional, debemos utilizar algoritmos capaces de abarcar el problema de forma práctica, desde el punto de vista de los resultados, por eso debemos preguntarnos

¿Cuáles son los criterios de desempeño a utilizar en la evaluación de los algoritmos?

1.2.5.2. Aplicación de algoritmos

Después de revisarse las tendencias y los algoritmos en la literatura, se deben escoger los algoritmos que se van utilizar para solucionar el problema en estudio, es ahí donde surge el interrogante:

¿Cómo usar los algoritmos seleccionados al problema de corte bidimensional tipo guillotina?

1.2.5.2. Medición de resultados

Es preciso realizar las corridas del experimento para así, con base en los criterios de evaluación establecidos, poder resolver la pregunta:

¿Cuáles son los resultados de los algoritmos usados?

1.2.5.3. Comparación

Luego de adquirir los resultados de las corridas experimentales y de realizar el análisis de resultado pertinente, se debe responder la siguiente pregunta

¿Cuál es el algoritmo que arroja mejores resultados de acuerdo con los criterios de evaluación establecidos?

1.3. OBJETIVOS

1.3.1. OBEJTIVO GENERAL

Adaptar un algoritmo para el problema de corte bidimensional tipo guillotina que cumpla con las restricciones planteadas en la literatura que permita disminuir desperdicios de láminas de acero

1.3.2. OBJETIVOS ESPECÍFICOS

- Escoger el modelo matemático para abordar el problema de corte bidimensional tipo guillotina.
- Seleccionar los métodos de solución del problema de corte bidimensional tipo guillotina y los diferentes enfoques utilizados para estudiarlo a través de una revisión del estado del arte.
- Emplear los algoritmos seleccionados para la solución del problema de corte bidimensional tipo guillotina.
- Medir los resultados y desempeños computacionales de los algoritmos aplicados al problema.
- Comparar mediante un análisis de resultado, el desempeño de los algoritmos tratados
- Elegir el algoritmo que revele mejores resultados del problema de corte bidimensional tipo guillotina

1.4. JUSTIFICACIÓN

El problema del corte bidimensional tipo guillotina, es un problema especialmente relevante en multitud de industria como metalúrgicas, papeleras, madereras, textiles, etc. Diseñar o adaptar algoritmos eficientes para este problema es de gran importancia para estas industrias por el aumento de rendimiento y competitividad que se obtendría al maximizar el beneficio de los cortes realizados. Básicamente el problema del corte bidimensional tipo guillotina consiste en dividir una lámina en segmentos rectangulares donde se deben ubicar piezas de tamaño y valor dado, de manera que se maximice la suma de los valores de las piezas cortada.(Pelález & Red, n.d.)

La optimización de los patrones de corte es importante con el fin de reducir el total de residuos generados por las operaciones correspondientes. (algoritmo de guillotina)

Por lo general, algunas empresas aún desconocen técnicas de optimización que les permita planificar como cortar su materia prima de tal manera que minimicen el desperdicio, lo que los lleva a incurrir en planificaciones basados exclusivamente en la intuición, Esto en muchos de los casos hace que se incurran en costos mayores de lo necesario en la adquisición de la materia prima. (Santis, Sthefany, Cuesta, Guillermo, & Palma, n.d.)

Uno de los grandes objetivos de las industrias manufactureras es la optimización del uso de sus materias primas, debido a que representan un alto porcentaje del costo de su producto terminado. El corte de materiales efectuado en sus procesos es fundamental para poder ser competitivas. Esto hace que el problema de corte bidimensional (2CP, Two-dimensional Cutting Problem) tenga una amplia gama de aplicaciones en industrias

dedicadas al corte de piezas de piel, tela, plástico, vidrio, metal y madera, por mencionar algunos. (Santis et al., n.d.)

En un ambiente altamente competitivo, el problema de corte de guillotina bidimensional es un elemento clave en la reducción de costos. Este problema tiene una amplia gama de aplicaciones en industrias cuyos procesos de corte de materiales se realizan con máquinas que sólo permiten cortes de un extremo a otro.(Santis et al., n.d.)

En este trabajo es adaptado un algoritmo para cortes tipo guillotina en 2D para láminas de acero usando metaheurísticas que permitan realizar la planificación de patrones de corte, de tal manera que se satisfaga una demanda preestablecida minimizando el número de áreas a utilizar reduciendo así desperdicios de láminas de acero, costos de producción, tiempo, aumentar el beneficio de los cortes realizados y elevar la competitividad de las empresas dedicadas al corte bidimensional tipo guillotina de láminas de acero.

Dentro de este proceso de corte se considerarán las posibles combinaciones de los requerimientos; debido al alto grado de combinaciones que se obtiene, presenta un alto costo de procesamiento y consumo de memoria principal de la computadora. En este sentido el trabajo permitirá construir patrones o esquemas de orden de complejidad lineal.(Rodriguez, Mauricio, & Rodríguez, 2015)

CAPITULO 2: MARCO DE REFERENCIA TEÓRICO GENERAL

2.1. MARCO TEÓRICO

2.1.1. CORTE Y EMPAQUETAMIENTO (C&P)

Los problemas de corte y empaquetado son una familia de problemas de optimización combinatoria que han sido ampliamente estudiados. Estos problemas nacen porque en muchas industrias de producción se trabaja con grandes hojas de material o rollos de materia prima, las cuales deben ser cortadas para obtener piezas más pequeñas, o incluso, en el caso del empaquetado de tres dimensiones, se debe determinar la posición de un conjunto de elementos para ser transportados o almacenados.

Los métodos de corte y empaquetado producen patrones que no son más que combinaciones geométricas de pequeñas piezas colocadas en objetos de mayor tamaño. Las áreas restantes de los objetos más grandes que quedan después de haber colocado todas las piezas pequeñas se les llama desperdicio. Especialmente en los problemas de corte, los objetos grandes son de materiales sólidos los cuales son cortados para obtener pequeñas piezas. Los materiales más usuales son acero, vidrio, papel, madera, piel, textiles, plástico y pasta. En los asuntos de problemas de carga y empaquetado, consiste en que se tiene un conjunto de objetos grandes vacíos, como contenedores, coches o furgones, cubos, palés, camiones, etc. El empaquetado de las pequeñas piezas en los objetos grandes se considera como cortes del espacio vacío en partes, algunas de las cuales están ocupadas por pequeñas piezas.

En los problemas de cortes, los elementos pequeños son una lista de piezas demandadas, en cambio los objetos de grandes dimensiones se conocen como stock o materia prima. Los compradores solicitan piezas, y las fabricas o industrias deben producir el conjunto

de piezas demandadas por medio del uso de la materia prima disponible (objetos grandes) para satisfacer a sus clientes. Al momento de tratar con los problemas de empaquetamiento, las piezas son los elementos que se deben cargar en los contenedores los cuales son el objeto grande, para su transporte o almacenamiento. (Delgadillo-Avila, 2007)

2.1.1.1. Estructura de un problema de corte

Para el problema de corte su estructura puede ser declarada de la siguiente manera:

- 1) Hay dos grupos de datos básicos donde los elementos son cuerpos geométricos de formas definidas en una o más dimensiones.
 - a) Grandes objetos o stock (bloques) de material (varillas, láminas, planchas) de tamaño fijo o variable que tienen que ser cortados según los pedidos de las piezas para satisfacer las necesidades del cliente, y
 - b) Un conjunto de ordenes requiriendo la producción de ciertas piezas (ítems), en general de diferentes tamaños, formas (figuras) y en determinadas cantidades.
- 2) Un proceso de corte que determina patrones (esquemas) estos como resultado de combinaciones geométricas de las piezas en los objetos, siguiendo:
 - a) Algunas especificaciones del material y de la tecnología del corte, restringido al número de procesos de cortes factibles, y
 - b) Ciertos objetivos a alcanzarse.

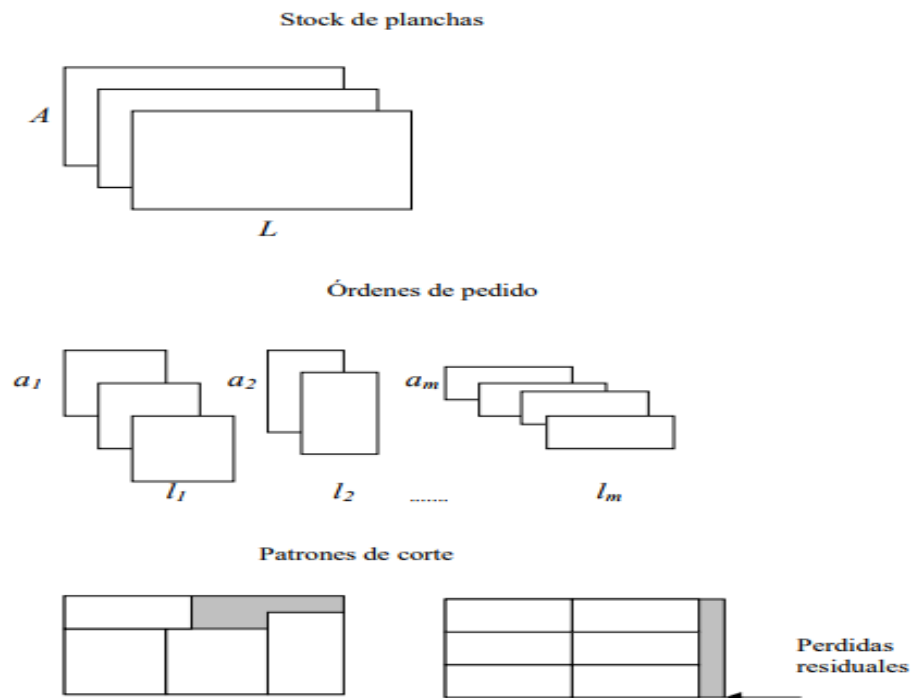
El desperdicio es obtenido cuando ya no es posible colocar piezas que se vayan a cortar en las áreas sobrante del objeto grande que se esté utilizando.

Para entender de mejor forma la estructura del problema de corte de dos dimensiones el cual se consigue al tratar de cortar láminas de vidrio, cartón, metal y vidrio, se muestra a continuación un ejemplo:

-Hay un stock ilimitado de láminas de tamaño $L \times A$, y una orden de pedido de m pequeñas piezas de dimensiones $l_1 \times a_1, l_2 \times a_2, \dots, l_m \times a_m$ tal que $l_i \leq L$, $a_i \leq A$ para todo $i=1,2,\dots,m$ que tendrá que ser satisfecha por el corte apropiado de las láminas.

-El proceso de construcción de los patrones de corte (órdenes para la producción de las piezas) obedece a objetivos y restricciones, que en este tema consisten en utilizar la menor cantidad de láminas y satisfacer las órdenes de las piezas.

Ilustración 1: Estructura general del problema de corte bidimensional



Fuente: (Delgadillo, 2007)

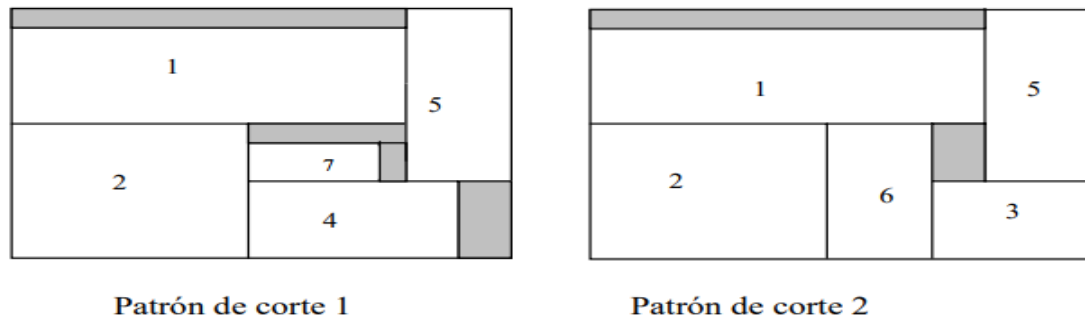
2.1.2. DEFINICIÓN DEL PROBLEMA DE CORTE DE DOS DIMENSIONES

Se encuentra con un stock de láminas de dimensiones $L \times A$, deben ser cortadas en piezas pequeñas de tamaño $l_i \times a_i$, suministrando N_i piezas. Las N_i piezas deben ser atendida por un número cualquiera de láminas.

El corte se elabora de la siguiente manera: es seleccionado un cierto número de patrones de corte rectangular, donde cada patrón (modelo) es definido como la forma de colocar las piezas $l_i \times a_i$ dentro de la lámina $L \times A$. El k -ésimo patrón de corte describe la manera de como la lámina deberá ser cortada para obtener las piezas de acuerdo a la demanda, y donde x_k es la cantidad de veces que el k -ésimo patrón de corte se utiliza.

El objetivo del problema de corte en dos dimensiones es atender las demandas usando el menor número de lámina.

Ilustración 2: Patrones de corte en 2 dimensiones



Fuente: (Delgadillo, 2007)

El problema de corte en dos dimensiones es formulado de la siguiente manera

(2DCSP)

$$\text{Min } 1 \cdot x$$

$$\text{s. a } Ax \geq N$$

$$x \geq 0$$

Donde:

$x = (x_1, x_2, \dots, x_k, \dots)$ es un vector de las variables x_k , una para cada columna de la matriz A , donde x_k es la cantidad de veces que el k -ésimo patrón de corte es usado.

$1 = (1, \dots, 1)$ es un vector fila de elementos todos iguales a uno.

$N = (N_1, \dots, N_n)$ es el vector columna de las demandas

A = Matriz de n filas cuyas columnas con formato (a_1, a_2, \dots, a_n) son los posibles modelos de corte rectangular del rectángulo $L \times A$.

a_i es el número de rectángulos $l_i \times a_i$ ($i = 1, \dots, n$) que ocurre en el modelo

De la Ilustración 2 Se puede notar que el patrón de corte 1 muestra que las piezas: 1, 2, 4, 5 y 7 con tamaños $(l_1 \times a_1)$, $(l_2 \times a_2)$, $(l_4 \times a_4)$, $(l_5 \times a_5)$, $(l_7 \times a_7)$ respectivamente han sido colocadas en la lámina $L \times A$. Si $n=8$, para un problema con este posible patrón de corte, una de las columnas de A será $(1, 1, 0, 1, 1, 0, 1, 0)$, y para el patrón de corte 2 como las piezas colocadas son: 1, 2, 3, 5 y 6, la columna de A asociada a este modelo será $(1, 1, 1, 0, 1, 1, 0, 0)$.

Si el objetivo que persigue el problema de corte en dos dimensiones es determinar un patrón de corte apropiado para cumplir con la demanda por medio del stock de láminas minimizando el desperdicio se le llama **problema del desperdicio por corte (The trim-loss problema)**, el cual se formula de la siguiente manera:

$$\min LxA - \sum_{i=1}^n x_i l_i a_i$$

$$s.a \quad 0 \leq x_i \leq b_i, \quad 1 \leq i \leq n$$

$$x_i \text{ entero}$$

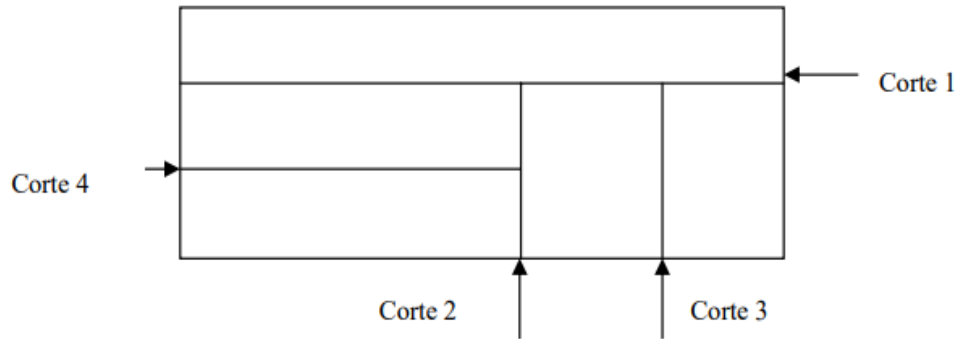
x_i , representa el número de veces que la pieza i es utilizada en el patrón de corte.

2.1.3. EL PROBLEMA DE CORTE POR GUILLOTINA

En aquellos problemas de corte en dos dimensiones que se utilizan ciertas máquinas que cortan el material de un extremo de la lámina al otro extremo opuesto, como por ejemplo la guillotina, la cual es utilizada para cortar material de acero y ya que fundamentalmente el contorno de las láminas y de las piezas son rectangulares o cuadrangulares; en este caso los patrones de corte son ortogonales. (Silva, Alvelos, & Valério de Carvalho, 2010) La complejidad de estos problemas depende del número de cambios de dirección de los cortes (**periodos**) y del número de cortes paralelos que se efectúa por períodos.

A continuación, se muestra un ejemplo de corte en periodos en la Ilustración 3, los cortes son enumerados en el orden en que se realizan.

Ilustración 3: Modelo de corte 2D por guillotina



Fuente: (Delgadillo, 2007)

Es posible definir cortes por guillotina en n periodos de forma inductiva como un estilo de corte en el que es considerado cada tira como una nueva lámina en la que es posible efectuar un corte por guillotina en $n-1$ periodos.

En (Furini, Malaguti, Medina Durán, Persiani, & Toth, 2012) se muestra un ejemplo de cortes de guillotina de dos etapas en un contenedor rectangular donde: los cortes de primera etapa (horizontal) se representan con una línea delgada y cortes de recorte (horizontal) con una línea de puntos. El área de desecho está representada por una superficie discontinua. Y se considera el caso en que los artículos pueden rotar ortogonalmente.

Ilustración 4: Corte de guillotina de dos etapas



Fuente: (Furini et al., 2012)

2.1.4. CLASIFICACIÓN DEL PROBLEMA DE CORTE

A continuación, se mostrará cómo está clasificado el problema de corte para este trabajo.

- **Clasificación por dimensión**

Ya que los patrones de corte son muy importantes en esta clase de problemas, así como también su naturaleza de combinaciones geométricas, lo que conlleva a que de los problemas C&E pertenecen al campo de la Geometría Combinatoria.

En este trabajo la dimensión es espacial debido a que las láminas y las piezas están definidas por dos dimensiones del espacio euclidiano.

- **Clasificación por la herramienta de corte**

Para diferenciar los problemas de corte bidimensional un factor muy importante es la herramienta que se utiliza para realizar el corte de las piezas demandadas.

En este trabajo la herramienta a utilizar es la guillotina lo cual aquí el problema es llamado corte de guillotina. Estos cortes son ortogonales de lado a lado (paralelos a los lados de la lámina).

- **Clasificación por el contorno de las piezas**

Por medio del contorno o bordes de los cortes (piezas) demandadas que deben ser atendidas el problema de corte bidimensional en este trabajo es un corte rectangular o regular.

- **Clasificación por características y tipos**

(Dyckhoff, 1990) muestra una clasificación más sistematizada referente a los componentes de la estructura del problema de corte y sus características geométricas y combinatorias, entre otras. A continuación, son mostradas.

❖ **Dimensionalidad:** aquí se captura el número mínimo (1, 2, 3, $n > 3$) de dimensiones geométricas que son necesarias para describir los diseños requeridos (patrones) por completo.

❖ **Tipo de asignación de objetos pequeños a objetos grandes:** Aquí se distinguen dos casos, los cuales son:

- **B:** del alemán “Berladeproblem”, este quiere decir que todas las láminas deben usarse, por lo cual se debe asignar una selección de piezas a las láminas.
- **V:** del alemán “Verladeproblem”, aquí todas las piezas deben asignarse a una selección de láminas

❖ **Surtido de las láminas:** Este tipo se representa de la siguiente manera

- **O:** una lámina
- **I:** varias laminas, pero idénticas
- **D:** varias láminas, pero diferentes

❖ **Surtido de las piezas:** se representan así

- **F:** se distinguen los surtidos que consiste en pocas piezas
- **M:** hay muchas piezas con figuras diferentes
- **R:** muchas piezas de relativamente pocas figuras diferentes
- **C:** figuras congruentes

De acuerdo a estos criterios de Dyckhoff el problema que se estudia en este trabajo es de tipo 2/V/I/R, debido a que se trata de un problema en dos dimensiones (láminas rectangulares), las piezas deben asignarse a un conjunto de láminas, se utilizan varias láminas, pero todas de igual dimensión y por ultimo las piezas son congruentes.

Además de estos criterios dados por Dyckhoff son introducidos otros y nuevas propiedades potencialmente importantes en (Wäscher, Haußner, & Schumann, 2007), a continuación, serán mostradas, específicamente las utilizadas para el problema de este trabajo:

- **Tipo de tarea**

Se presenta la siguiente situación básicas;

- **Minimización de entrada (valor):** Donde un determinado conjunto de piezas debe asignarse a un conjunto de láminas. Aquí el conjunto de láminas es suficiente para acomodar todas las piezas. Todas las piezas se colocarán a una selección de láminas, minimizando el número de láminas a utilizar y así evitar mucho desperdicio. En esta situación no hay problema de selección con respecto a las piezas.

- **Surtido de artículos pequeños**

Con respecto al surtido de las piezas en este problema es distinguido un *surtido débilmente heterogéneo* donde las piezas se pueden agrupar en relativamente pocas clases (en relación con el número total de elementos), para los cuales los elementos son idénticos con respecto a la forma y tamaño. Por definición, los artículos pequeños de idéntica forma y tamaño que requieren diferentes orientaciones se tratan como diferentes tipos de artículos. La

demanda de cada tipo de elemento es relativamente grande, y puede o no estar limitada por un límite superior. En forma simplificada se trata de muchas piezas de relativamente pocas formas diferentes (no congruentes).

- **Surtido de objetos grandes**

De acuerdo a este criterio en este trabajo se utilizan varios objetos grandes, porque consta de láminas con dimensiones fijas, es decir las láminas son idénticas y existe un débil y surtido muy heterogéneo.

- **Forma de artículos pequeños**

En este trabajo la forma de las piezas es regular (rectángulo) las cuales deben ser presentadas de forma ortogonal.

2.1.4.1. TIPO DE PROBLEMA BÁSICO

El tipo básico de problemas de C&P se desarrollan combinando los dos criterios “tipo de asignación y “surtido de elementos pequeños”. En este trabajo es utilizado el tipo de asignación de minimización de entrada y el surtido de elementos pequeños es un *surtido débilmente heterogéneo* por lo tanto el tipo básico es el problema de corte de stock (cutting stock problem) el cual consiste en que un surtido débilmente heterogéneo de artículos pequeños esté completamente colocado en una selección de objetos grandes, minimizando el desperdicio de estos. Los objetos grandes pueden ser idénticos, pero también podrían ser un surtido débil o de tamaños fuertemente heterogéneos.

2.1.4.2. TIPO DE PROBLEMA INTERMEDIO

Debido a que la característica de las láminas es que todas son de igual dimensión y que las piezas son un surtido débilmente heterogéneo, en este trabajo se está tratando con un

problema de corte de stock del tamaño de una sola lámina (Single Stock Size Cutting Stock Problem -SSSCSP-). el SSSCSP bidimensional (problema de stock de corte de un solo stock), es el problema en el cual m tipos de artículos rectangulares con demandas específicas se producen completamente a partir de placas del mismo tamaño $L \times W$ (longitud \times ancho), de modo que la entrada (número de placas utilizadas, simplemente "recuento de placas") se minimiza. Los tipos de elementos se caracterizan por longitudes (l_1, \dots, l_m) , anchos (w_1, \dots, w_m) y demandas (d_1, \dots, d_m) . Una solución del SSSCSP es un plan de corte que contiene un conjunto de diferentes patrones de corte, cada uno de los cuales tiene diseño y frecuencia especificados. (Cui & Zhao, 2013)

En el problema clásico bidimensional de stock de corte (Gilmore & Gomory, 1965), un conjunto de rectángulos débilmente heterogéneos tiene que ser cortado de placas de un tamaño específico (largo y ancho). En el cual se debe minimizar el número o el valor de los objetos grandes necesarios.

2.1.5. PROBLEMAS DE CORTES BIDIMENSIONALES NO LIMITADOS

(U2DCP)

En el U2DCP se toma una lámina rectangular de dimensiones L Y A para sacar piezas rectangulares de n tipos $\{s_1, s_2, s_3, \dots, s_n\}$, donde cada uno tiene sus dimensiones (l_i, a_i) y el beneficio π_i . Las piezas se toman de la lámina, sin intercalación, por medio de una sucesión de cortes rectos horizontales y verticales de guillotina (de un extremo al otro extremo de la lámina), ósea cortes que fraccionan la lámina en dos partes. Se permite rotación de la pieza.

Un conjunto de cortes tipo guillotina se conoce como esquema de corte, Cada conjunto corresponde a un patrón de corte, es decir, un vector entero n -dimensional

$((d_1, d_2, d_3, \dots, d_n))$ donde d_i representa el número de piezas de tipo s_i extraídas de la lámina. No existe límite en los valores b_i .

Si los beneficios π_i son iguales a las áreas de las piezas, el problema se denomina ponderado y consiste en encontrar la solución óptima que minimiza la pérdida de recorte, de lo contrario se denomina ponderada y consta de hallar la solución óptima que maximiza el beneficio asociado al conjunto de piezas extraídas. (Gilmore P. C. and Gomory R. E., 1961)

2.1.6. ALGORITMIA

Un algoritmo es un conjunto de instrucciones donde se puntualiza la secuencia ordenada de instrucciones que se deben realizar para solucionar un problema. En otras palabras, un algoritmo es un procedimiento o técnica para la resolución de un problema. Un algoritmo es independiente tanto del lenguaje de programación en que se formule como del ordenador en el que se produzca.

Por lo tanto, podemos ver que su objetivo es hallar la solución de un problema y debe cumplir una serie de características como lo es el tener un orden lógico y comprender un inicio y un final. Consiguientemente se debe pensar en un algoritmo como una receta a seguir para resolver un problema determinado, normalmente a partir de una información o de datos de partida o de entrada que puede variar (Vancells Flotats, 2002).

Ejemplo: Una receta para hacer un pastel (descrita de forma precisa y sin decisiones subjetivas) puede ser calificada como un algoritmo porque indica los pasos que se deben realizar para resolver el problema (pastel) y es independiente tanto del idioma en que se escriba como del repostero que la ejecute.

Un algoritmo se caracteriza por:

- Debe ser claro y puntual (sin imprecisiones), e indicar el orden de ejecución de cada paso
- Debe ser imaginable. Si se aplica partiendo de la misma situación inicial, se debe obtener siempre el mismo resultado.
- Debe ser limitado. El algoritmo debe finalizar en un algún instante (debe tener un número limitado de pasos).

2.1.7. COMPLEJIDAD ALGORITMICA: PROBLEMAS NP-DURO

Matemáticamente, los problemas se describen atendiendo a la dificultad que entraña su resolución por un ordenador. Es por esto que se han definido varias clases de problemas, entre las que se recalcan las clases P, NP, NP-completo Y NP-duro.

La idea intuitiva de problema “difícil de resolver” queda reflejada en el término científico NP-hard utilizado en el contexto de la complejidad algorítmica. En términos coloquiales podemos decir que un problema de optimización difícil es aquel para el que no podemos garantizar el encontrar la mejor solución posible en un tiempo razonable. La existencia de una gran cantidad y variedad de problemas difíciles, que aparecen en la práctica y que necesitan ser resueltos de forma eficiente, impulso el desarrollo de procedimientos eficientes para encontrar buenas soluciones, aunque no fueran las óptimas.

2.1.8. MÉTODOS DE SOLUCIÓN DEL PROBLEMA BIDIMENSIONAL TIPO

GUILLOTINA

2.1.8.1. HEURÍSTICAS

Debido a que la cantidad de tiempo para encontrar una solución exacta a un problema es completamente inaceptable es necesario hacer uso de heurísticas o métodos aproximados que nos ayuden a encontrar un resultado aproximado (de calidad), pero con un tiempo razonable de búsqueda.

El término heurística se debe al matemático G. Polya quien lo empleo por primera vez en su libro *How to solve it*. La más clara definición que se tiene sobre heurística es dada por (Zanakis & Evans, 1981) que nos dice que la heurística son los ‘procedimientos simples, a menudo basados en el sentido común, que se supone que obtendrán una buena solución (no necesariamente óptima) a problemas difíciles de un modo sencillo y rápido’.

Los métodos heurísticos tienen una desventaja la cual consiste en que tienen una inhabilidad de escapar de óptimos locales ya que estos algoritmos no constan con unos componentes que les permitan seguir en busca del óptimo debido a que se quedan atrapados en óptimos locales. Es por esto que se meten otros algoritmos de indagación más inteligentes (llamados Metaheurísticas) que impiden, en la medida de lo posible, este tipo de problemas. Este tipo de algoritmo son programaciones de alto nivel que guían a métodos heurísticos conocidos, impidiendo que estos queden atrapados en óptimos locales.

Las heurísticas se utilizan cuando no hay un método exacto que le dé solución a un problema en particular o cuando si existen, pero no arrojan la solución óptima en un

tiempo prudente y también cuando existen condiciones de tiempo o como paso intermedio para obtener una solución inicial para la aplicación de otra técnica.

Algunos autores proponen la siguiente clasificación de métodos de resolución mediante heurísticos:

- Métodos constructivos, estos tratan de construir una solución especificando diferentes partes de ella en pasos continuados.
- Métodos de descomposición, este método divide el problema en problemas más pequeños cuya solución se obtiene a partir de cada solución de los más pequeños
- Métodos de reducción, consiste en identificar una cualidad de la solución la cual permita hacer más sencilla el procedimiento del problema
- Métodos de manipulación del modelo, este tipo de métodos logran una solución del problema original a través de la solución de otro problema simplificado
- Métodos de búsqueda por entornos, consta de comenzar de una solución inicial a la cual se le va haciendo innovaciones en encadenadas iteraciones para lograr una solución final. En cada iteración existe un conjunto de soluciones vecinas aspirantes a ser la solución del proceso. En este grupo se encuadran las técnicas metaheurísticas. (García Sánchez, 2013)

2.1.8.2. METAHEURISTICAS

Son procedimientos de búsqueda que tampoco garantiza dar una solución óptima del problema que se esté tratando, pero que a diferencia de las heurísticas están tratando de escapar de óptimos locales orientando la búsqueda en cada instante dependiendo del progreso del proceso de búsqueda.

La lógica de las técnicas metaheurísticas consiste en que el punto de inicio es una solución (o conjunto de soluciones) que normalmente no es óptima. De la cual se adquieren otras similares, de estas se elige una que satisface cierto criterio, a partir de la cual comienza un nuevo proceso. Este proceso es detenido cuando se cumpla cierto parámetro de parada.

Una de las definiciones más descriptivas es la presentada por (Osman, Ct, & Kelly, 1996) y es la siguiente: ‘Las metaheurísticas son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria en los que los heurísticos clásicos no son efectivos. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos’.

A medida que va pasando el tiempo se ha podido dar cuenta que por medio de la metaheurística se han resuelto una cantidad de problemas que inicialmente parecían inaccesible.

Entre las metaheurísticas que más se conocen son los métodos tan populares como optimización por colonias de hormigas (ACO), algoritmos evolutivos (EA), donde se incluyen los algoritmos genéticos(GA) y los algoritmos meméticos (MA), procedimientos de búsqueda miope (constructiva, voraz o ávida), aleatorizados y adaptivos (GRASP), búsqueda local iterativa (ILS), re-encadenamiento de trayectorias (PR), recocido simulado(SA), búsqueda dispersa(SS) y búsqueda tabú(TS).

2.2. ESTADO DEL ARTE

Los problemas de corte y empaquetado se remontan a 1939 con Kantorovich y 1949 con Brooks. Sin embargo, Gilmore y Gomory en 1961 presentaron la primera publicación relevante en esta área. Sweeney y Paternoster en 1992 hacen una recopilación y clasificación de las publicaciones hechas entre 1940 y 1990 en base a las dimensiones, tipo del problema y tipo de aproximación empleada en la solución.(Piezas et al., 2014)

Desde los años 60 se han diseñado distintos tipos de algoritmos para solucionar el problema de corte tipo guillotina. Para ello se han utilizado desde técnicas tradicionales como programación dinámica o lineal, métodos heurísticos, y finalmente técnicas de inteligencia artificial.(Peláez & Red, n.d.)

En este contexto para solucionar el problema de corte tipo guillotina Gilmore y Gomory inician esta investigación en la década de los 60 con los métodos exactos y constituyen la primera etapa de investigación en el área de cortar y llenar. Usaron técnicas de programación lineal para solucionar problemas de optimización unidimensionales. Se han realizado mejoras y modificaciones a lo largo del tiempo. Sin embargo, en general los métodos exactos son buenos sólo para problemas pequeños y medianos, debido a los altos requerimientos computacionales (Rodriguez et al., 2015) . Mas adelante Gilmore y Gomory en 1965 y 1966 proponen un algoritmo recursivo exacto sobre la base de la programación dinámica para resolver el problema. Este algoritmo es aplicable a las versiones con y sin pesos. Luego (Herz, 1972) propone un método de búsqueda recursiva de árbol, que es más eficaz que el algoritmo de Gilmore y Gomory para el problema sin pesos, pero no se aplica a los casos con pesos.

Existen dos técnicas generales usadas para resolver los problemas restringidos: *top-down* y *bottom-up* (*de arriba hacia abajo* y *de abajo hacia arriba*). (Christofides & Whitlock, 1977) propusieron originalmente el enfoque de *top-down* (*de arriba hacia abajo*); el otro requiere una gran cantidad de memoria, razón por la cual la implementación de un algoritmo de este tipo es poco atractivo. (Toro-ocampo, 2010)

Por otro lado, los enfoques heurísticos más documentados son el (LB) inferior - izquierdo y los métodos de (LBF) mejor - inferior - izquierdo. En 1980, Brown revisó el límite inferior de la estrategia de colocación inferior - izquierda y demostró que hay grupos de rectángulos para los que es imposible producir la solución óptima. Ésta es la desventaja principal de la heurística de colocación LB. El segundo método, LBF, es una versión modificada de la heurística de colocación inferior - izquierda. El método LBF, puede llenar espacios libres usando rectángulos en el empaquetamiento, mientras que en LB este espacio nunca será llenado (incluso si un rectángulo posterior quedara bien) y por lo tanto el espacio puede ser considerado desperdicio. El algoritmo LBF es perjudicado por su peor complejidad de tiempo de $O(N^3)$, por consiguiente, el tiempo de ejecución puede ser muy grande en problemas de mayor tamaño. (Rodríguez et al., 2015)

Otra de las técnicas que aún no ha sido muy difundida, es el agrupamiento. Un acercamiento de esta estrategia fue presentada (Kröger, 1995) quien en 1995 usó algoritmos genéticos para el embalaje de recipiente de guillotina y lanzó la idea de "Meta-rectángulos", que son agrupaciones de rectángulos que pueden empaquetar bien juntos y pueden ser tratado como un único objeto solo en futuras repeticiones de búsqueda. (Rodríguez et al., 2015)

Más tarde (Faggioli & Bentivoglio, 1998), investigaron los métodos tanto heurísticos como exactos en un enfoque de tres fases. Esto suponía producir una buena solución de arranque en primer lugar con un algoritmo goloso, mejorar con búsqueda tabú y luego usar procedimientos de enumeración implícitos exactos. El método es bueno, pero involucra un esfuerzo computacional muy grande para 40 o más piezas.

Por otro lado, con el propósito de mejorar el empaquetamiento y reducir el desperdicio, se propuso la rotación de las piezas que van a ser encajadas en las láminas. Una muestra de esta estrategia se da en el año 2000, cuando Nagao, Sawa, Shigehiro y Shirakawa, abordan el problema de empaquetamiento de rectángulos desde la perspectiva de diseño de chips VLSI. Este tipo de problema tiene la propiedad que el tamaño de la lámina no es pre-especificada, pero su área rectangular debe estar minimizada (un factor importante en la Microelectrónica). Sin rotación se obtiene una solución 2.91% superior al óptimo, con rotación la solución mejora al 1.65% superior al óptimo.

Más tarde Feo y Resende desarrollan el algoritmo GRASP, el cual es un procedimiento iterativo que combina una fase constructiva y una fase de mejora. En la fase constructiva, una solución es construida paso a paso, añadiendo elementos a una solución parcial. Para escoger el elemento para ser añadido, una función golosa, que dinámicamente es adaptada como la solución parcial es construida. Sin embargo, la selección del elemento no es determinista, pero sujeta a una randomización del proceso. De este modo, cuando repetimos el proceso podemos obtener soluciones diferentes. Después de cada fase constructiva, la fase de mejora, por lo general consiste en una búsqueda simple local, que trata de substituir algunos elementos de la solución, que están allí como el resultado de la ordenación aleatoria, con otros, produciendo así una mejor solución total.

Una propuesta de este método es la desarrollada por Mauricio y Rivera en el año 2002 y 2003. Los algoritmos propuestos de construcción Grasp son basados en los algoritmos golosos FFD y BFD de Mauricio, pero se diferencian en el criterio de selección, y no se utiliza el criterio decreciente, por cuanto no existe un ordenamiento para atender los requerimientos, sino que se efectúa por una selección aleatoria desde RCL.

Es posible introducir variantes mediante el uso de parámetros de relajación. La segunda componente de un algoritmo GRASP es un algoritmo de búsqueda local, el cual tiene por objetivo, mejorar la construcción encontrada por el algoritmo de construcción GRASP. Si se utilizan adecuadamente los parámetros de relajación, permite diversificar las soluciones, lo que significa que puede obtenerse mejores soluciones que los algoritmos FFD y BFD. (Rodriguez et al., 2015)

Al mismo tiempo en (Onwubolu & Mutingi, 2003) 2003, propusieron un enfoque genético para el problema de embalaje de rectángulos de tipo guillotina. El enfoque pone un rectángulo inicial en primer lugar y luego busca un rectángulo que puede ser adjuntado al primero mientras mantiene restringida el corte de guillotina. Los cromosomas dentro del algoritmo genético representan las piezas que pueden ser adjuntados por pares y la orientación en la que deben ser puestas. Los autores informaron sobre una mejora de más de 60% de los problemas usados en trabajos anteriores.(Rodriguez et al., 2015)

Sin embargo, G, Kang y Seong también en 2003 plantearon un algoritmo que parte de una solución inicial de buena calidad y utiliza el algoritmo constructivo *bottom-up* como estrategia para generar las ramas y así disminuir el número de nodos por explorar. (Toro-ocampo, 2010)

Años más tarde específicamente en 2006 Bouriet *proponen* una nueva heurística para resolver el problema de corte rectangular de tamaño múltiple que minimiza razonablemente el desperdicio y los tiempos computacionales. Pero luego Toro y Granada en 2007 utilizan un algoritmo genético modificado para encontrar un patrón de corte rectangular tipo guillotina y sugieren una estrategia de codificación del problema basada en cortes por secciones. Así como también (Terashima-Marín, Farías-Zárate, Ross, & Valenzuela-Rendón, 2007) presentan dos modelos basados en computación evolutiva para producir hiper-heurísticas que resuelvan problemas de empaquetado bidimensional.

Enseguida en (Toro, Eliana, Rueda, & Augusto, 2007) presentan un algoritmo de búsqueda tabú para el problema de corte bidimensional tipo guillotina usando diferentes soluciones constructivas de inicio para observar su efecto sobre la solución encontrada después de varias iteraciones. En el 2008 (Yanasse & Morabito, 2008) presentan modelos lineales y no-lineales para generar un grupo de patrones de corte bidimensionales tipo guillotina con y sin restricciones incluyendo casos exactos y no exactos.

Estos modelos son útiles para la investigación y desarrollo de métodos de solución efectivos, exploración de estructuras particulares, modelos de descomposición y relajación, entre otros. También en ese mismo año (Clautiaux, Jouglet, Carlier, & Moukrim, 2008) presentan una nueva clase de coloreado de grados orientados, llamados grafos guillotina, que modelan patrones guillotina y proponen un algoritmo de reconocimiento para esos grafos.

Más tarde en 2009 Yaodong presenta un algoritmo para el problema de corte bidimensional restringido de objetos rectangulares que genera recursivamente un patrón óptimo de bloque uniforme y selecciona de manera óptima la primera tira de corte a partir

de la sub-placa actual. (Hifi, M'Hallah, & Saadi, 2009) por otro lado proponen un algoritmo exacto de ramificación y acotamiento y un algoritmo aproximado de dos etapas que usa esquemas de cota inferior y superior eficientes.

Por consiguiente, (Amossen & Pisinger, 2010) presentan una generalización de un algoritmo constructivo para el problema de empaçado multidimensional, con y sin restricciones de guillotina, basado en programación de restricciones. También (Morabito & Pureza, 2010) en este mismo año presentan un método heurístico para generar patrones restringidos de corte de guillotina de dos dimensiones, usando un espacio de relajación de una formulación de programación dinámica del problema y un procedimiento ascendente de optimización subgradiente. Así como también (Picardie, Verne, & Sciences, 2010) estudian un problema de corte de orientación fija bidimensional de dos etapas e investigan el uso de un algoritmo de búsqueda en haz paralelo para resolver el problema de manera aproximada.

Recientemente se ha popularizado la utilización de métodos híbridos en la solución de problemas de corte de guillotina de dos dimensiones que consisten en la combinación de metaheurísticas como recocido simulado, búsqueda tabú, algoritmos evolutivos, métodos exactos, entre otros. Es por esto que en (Aryanezhad, Fakhim Hashemi, Makui, & Javanshir, 2012) propone un nuevo método heurístico práctico y rápido para la solución de un problema de corte de dos dimensiones con elementos y placas rectangulares utilizando cortes de guillotina. En su implementación el usuario tiene acceso a información relacionada con la dimensión y número de cortes requeridos, la dimensión y número de piezas de desecho y el porcentaje de desperdicio producido. Más tarde Paret presenta un problema de corte de guillotina de dos dimensiones mediante un algoritmo de

dos etapas. Ellos usan un método de bandas de empaque para resolver un problema procedente de la industria del vidrio. El algoritmo propuesto genera niveles o bandas donde los elementos se seleccionan en base a la longitud de la tira considerando una rotación de 90° . (Piezas et al., 2014)

Por último, más reciente (Lu & Huang, 2015) se ha propuesto un algoritmo genético eficaz que incorpora un procedimiento de colocación novedoso, denominado algoritmo de espacio de esquina, y un modelo de programación de enteros mixtos para resolver el problema. Los objetivos clave son reducir los costos totales de producción y satisfacer los requisitos de los clientes.

2.2.1. HEURÍSTICAS APLICADAS AL PROBLEMA DE CORTE BIDIMENSIONAL TIPO GUILLOTINA

El problema de corte bidimensional fue presentado por Gilmore y Gomory. (Alvarez-Valdés, Parajón, & Tamarit, 2001) desarrollo y comparó varios métodos heurísticos, basados en la generación de columnas y el redondeo de la relajación de programación lineal, para resolver el problema de corte bidimensional. De aquí se presenta una heurística de agrupamiento secuencial (SHP). La cual se basa en el procedimiento secuencial (SHP) que genera cada siguiente patrón para cumplir con una parte de los elementos restantes y se repite hasta que todos los elementos se cumplan. El SGH ajusta los valores de los elementos de acuerdo con el método de corrección de valores secuenciales y utiliza una técnica de agrupación para seleccionar los elementos que se pueden utilizar para generar el siguiente patrón. El SGH se amplía para soportar la computación paralela. Los resultados computacionales indican que el SGH es potente tanto en la minimización de entrada como en la reducción de patrones. Luego (Cui & Zhao, 2013) presentan un

algoritmo heurístico para resolver el problema rectangular del stock de corte de tamaño de stock con patrones de dos etapas. Utiliza el método de generación de columnas para resolver los problemas residuales repetidamente, hasta que satisfagan las demandas de todos los elementos. Cada patrón se genera utilizando un procedimiento para el problema de colocación de objetos grande único restringido para garantizar la convergencia del algoritmo. (Furini et al., 2012) propuso un enfoque basado en la generación de columnas y una heurística de buceo, mejorando en los resultados computacionales obtenidos en (Cintra, Miyazawa, Wakabayashi, & Xavier, 2008) donde se presentó un algoritmo basado en la generación de columnas para este problema que utiliza el segundo algoritmo mencionado anteriormente donde se imponen patrones por etapas. Recientemente, (Malaguti, Medina Durán, & Toth, 2014) estudio una extensión del mundo real del problema donde el objetivo de minimizar una combinación ponderada de uso de material y el número de patrones diferentes en la solución. Propusieron procedimientos heurísticos y un algoritmo de ramificación y precio truncado.

2.2.2. METAHEURISTICA APLICADAS AL PROBLEMA BIDIMENSIONAL TIPO GUILLOTINA

Debido a que las metaheurísticas son estrategias para diseñar procedimientos heurísticos, una clasificación natural de meta heurísticas se establece en primer lugar en función del tipo de procedimiento a los que se refiere (Melián, Pérez, & Vega, 2003). Algunos de estos tipos fundamentales son:

- **Metaheurísticas de relajación:** Se refieren a los procedimientos de resolución de problemas que utilizan relajación del modelo original, y cuya solución facilita la solución del problema original.

- **Metaheurísticas constructivas:** Se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de las componentes que la forman.
- **Metaheurísticas de búsqueda:** Guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociados.
- **Metaheurísticas evolutivas:** Se enfocan a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.

En esta parte se muestra un recorrido del algoritmo genético ya que es la metaheurística que más ha sido estudiada en este campo de investigación como es el problema de corte bidimensional tipo guillotina.

2.2.2.1. ALGORITMOS GENÉTICOS (GA)

La aplicación de los algoritmos genéticos es ahora omnipresente en los estudios relacionados con los materiales y la fabricación. Algunos ejemplos típicos incluyen estudios sobre el trabajo mecánico de metales y su mecanizado, aleación desarrollo, fundición y solidificación, laminación en caliente y en frío. Aunque estas técnicas biológicamente inspiradas son muy nuevas en estas áreas de investigación, en los últimos años se aplicaron algoritmos genéticos para determinar el diseño optimizado de las partes rectangulares relacionadas con problemas de corte de metal. (Tiwari & Chakraborti, 2006)

Utilizaron un algoritmo genético multi-objetivo codificado en el árbol para estudiar tanto los casos de corte con guillotina como sin guillotina, utilizando una representación binaria de las variables, y se muestra para los casos conocidos que se obtienen las soluciones globalmente óptimas.

En (Michalewicz, Janikow, & Krawczyk, 1992) se propone utilizar algunas ideas de algoritmos genéticos para escanear el dominio del problema. Una solución factible a este problema se representa por medio de una cadena. A partir de un conjunto inicial de cadenas, se generan poblaciones sucesivas, todas ellas correspondientes a nuevas soluciones factibles. Después de un número finito de generaciones se encuentra una buena solución para este problema.

(Péter, 2017) presentan un nuevo método de solución para el problema de corte bidimensional, el cual se basa en una representación en árbol de las combinaciones de piezas y en un algoritmo genético adecuadamente adaptado, este se desarrolla para la resolución del problema restringido, cuando cada corte debe ser un corte guillotina, para desarrollarlo el primer paso es desarrollar una codificación adecuada de los patrones de corte. En este orden se utilizó una representación en árbol del patrón de corte, donde cada nodo representa una pieza que se corta. El método presentado puede reducir la carga computacional de todo el cálculo y da buenos resultados.

CAPITULO 3: METODOLOGÍA

El objetivo principal de este trabajo es minimizar el residuo o desperdicio de materiales que se generan en el proceso de corte generando así un incremento de la productividad y buscándole solución a un problema relevante en multitud de industrias tales como papeleras, textiles, madereras, metalúrgicas y vidrios.

El enfoque de la investigación es de tipo cuantitativo ya que se hará uso de herramientas estadísticas para la solución del problema de corte bidimensional tipo guillotina y cuyo tipo de estudio que se implementará en la realización de este proyecto es de carácter correlacional, pues se pretende comparar varios algoritmos que solucionen el problema de corte bidimensional tipo guillotina y entre ellos escoger el mejor, es decir el que arroje mejores resultados. Por tanto, la realización del proyecto investigativo será basado en técnicas específicas de recolección de la información, tales como: algoritmos que solucionen el problema de corte tipo guillotina 2D

De acuerdo con los objetivos planteados, el método de investigación que se utilizará durante el desarrollo de este proyecto de investigación es de carácter analítico, ya que se procederá a estudiar cada uno de los algoritmos que solucione este tipo de problemas en las industrias.

El método analítico en esta investigación permitirá, por medio de la revisión bibliográfica determinar cuál de todos los algoritmos ya propuestos brinda mejor resultado.

El interés de esta investigación se centra en la identificación de los algoritmos más convenientes para la solución del problema de corte bidimensional tipo guillotina cuando las láminas pueden ser giradas a 90° .

Luego de haberse estudiado el estado del arte se han seleccionados las técnicas metaheurísticas como las herramientas de solución a emplear, debido a su amplia utilización en el campo de la optimización y especialmente a su aplicación al problema de corte bidimensional tipo guillotina. Las metaheurísticas escogidas son Algoritmo Genético GA, Búsqueda Tabú y una metaheurística recientemente desarrollada por (Kaveh & Dardas, 2017) Llamado Optimización del Intercambio Térmico TEO. La escogencia de GA y Búsqueda Tabú es justificada porque han sido usados en este tipo de problema, lo que al contrario el TEO no ha sido utilizado en esta clase de problemas es por esto que se busca incursionar en su aplicación.

En el algoritmo GA se usó un operador de cruzamiento hallado en la literatura que demostró una mejor transmisión del material genético de encoding permutado y un operador de mutación, que por un diseño experimental hecho anteriormente en (Álvarez & Noble, 2013) fueron escogidos como mejores para su comparación con los demás algoritmos, en el caso del Búsqueda tabú se utiliza un operador de mutación más exactamente llamado swap y por último en TEO fueron usados dos operadores de mutación; el shift y swap

En último lugar se realizó un análisis de resultado para comparar por medio de tabla dinámica los resultados logrados de los diferentes algoritmos programados, de acuerdo a calidad de soluciones.

3.1. REPRESENTACIÓN DE LAS SOLUCIONES (ENCODING)

Para la representación apropiada de las soluciones maniobradas por los algoritmos se utilizó un encoding mixtos de tamaño $3n+1$ (donde n es el número de piezas a colocar) compuesto por cuatro partes básicas. La primera parte es un vector permutado de tamaño

n que muestra el orden en que serán colocadas las piezas en la lámina. La segunda parte es un vector binario de tamaño n que muestra la forma del corte, es decir, si el valor es 0 o 1 será un corte horizontal o vertical respectivamente. La tercera parte de tamaño n es un vector binario que representa la orientación de la pieza correspondiente, ya sea 1 para mantener la orientación original o 0 para girar 90° oportuna. Posteriormente, la última parte contendrá el valor de la función objetivo de la solución correspondiente.

Para mayor entendimiento se expone un ejemplo del encoding donde el número de piezas a colocar es $n=9$.

Ilustración 5: Ejemplo de encoding de una solución



Fuente: Autora

El anterior ejemplo muestra que la primera pieza a colocar será la N°3, esta se colocará con su orientación de entrada, sobre la esquina inferior izquierda de la lámina de acero con un corte horizontal, luego se ubicará la pieza N°8 con un corte vertical y con una rotación de 90° con respecto a su orientación original, luego se pondrá la pieza N°9 con un corte horizontal y manteniendo su orientación original y así sucesivamente hasta colocar las 9 piezas en la lámina.

3.2. DESCRIPCIÓN DE LOS ALGORITMOS

3.2.1. Algoritmo Genético (Genetic algorithm: GA)

La técnica metaheurística de los algoritmos genéticos fue ideada por Holland en 1975 y está inspirada en el proceso de adaptación de los seres vivos.

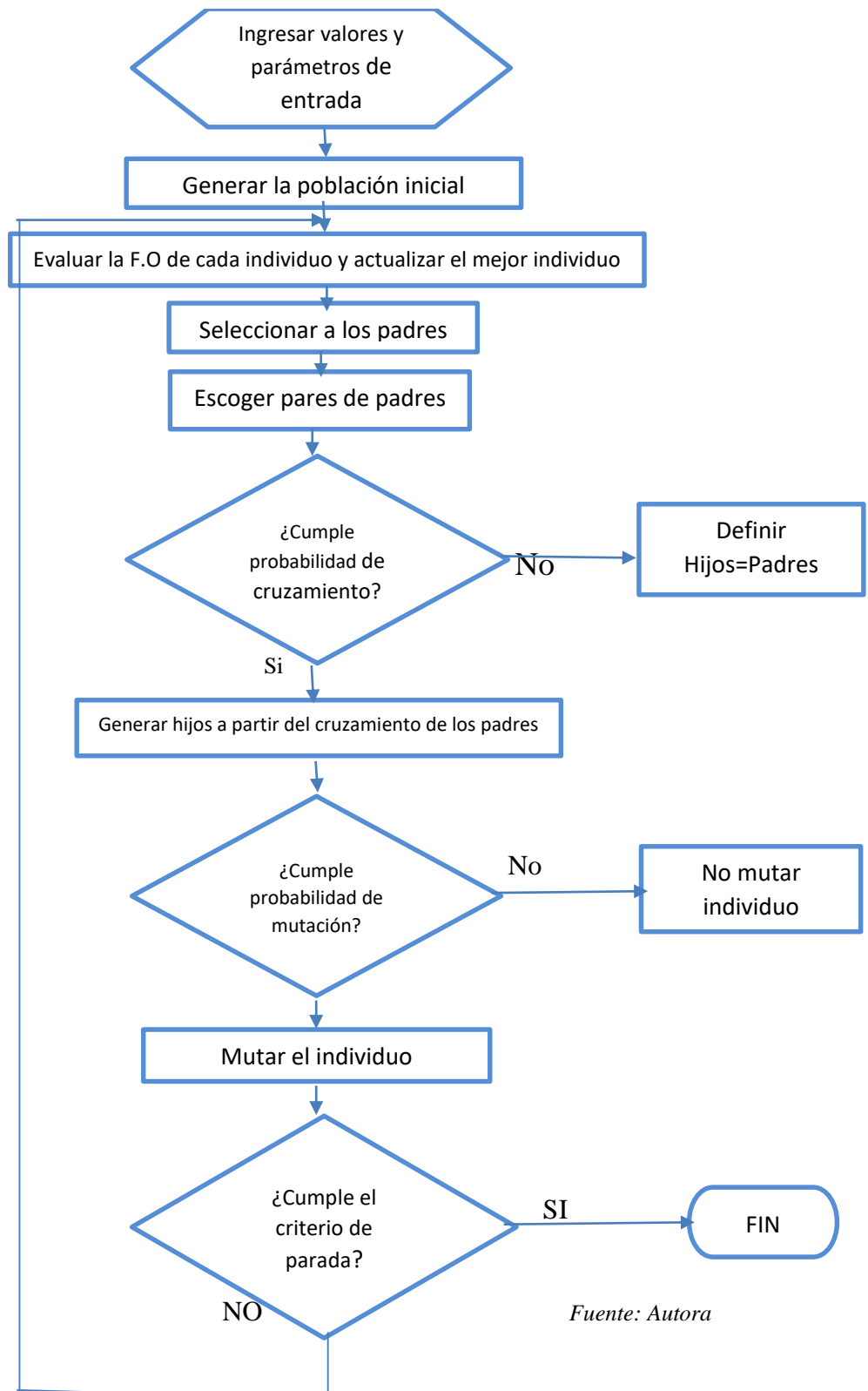
Este algoritmo consiste en que se parte de una población inicial (conjunto de soluciones) se realizan manipulaciones por las que se obtienen sucesivas poblaciones.

En cada iteración se realizan una serie de operaciones con los individuos de la población, las más comunes son: la selección, el cruce, la mutación y la inversión cuya aplicación permite obtener, típicamente, soluciones con mejores funciones de adaptación.

El algoritmo genético pertenece al grupo de las técnicas evolucionarias, las cuales son aquellas técnicas que en cada iteración dispone de un conjunto de soluciones a partir de las cuales obtienen un nuevo conjunto de soluciones. (García Sánchez, 2013)

A continuación, se explicará brevemente en que consiste esta clase de algoritmo. El primer paso es la generación de la población inicial de soluciones al problema establecido. Luego continua con la evaluación de dicha población, posterior a esto se hace la selección de los padres por medio de torneo, el siguiente paso es hacer el cruzamiento y mutación de las soluciones dentro de la población que tenga mejores resultados a criterio de la función objetivo. Luego del cruzamiento y mutación se eligen las soluciones que harán parte de la nueva población y se evalúa nuevamente esta nueva población resultante para continuar con el ciclo hasta que complete el criterio de parada. Esta explicación se puede ver mejor en el diagrama de flujo del algoritmo mostrado en la Ilustración 6:

Ilustración 6: Diagrama del Algoritmo Genético



Fuente: Autora

Para el problema expuesto en este trabajo y dadas las condiciones del encoding usado se explicarán los pasos expuesto en el diagrama con más detalle para una mayor ilustración y comprensión del algoritmo aplicado al problema.

1. La generación de la población inicial se realizó de manera aleatoria y se evalúa la función objetivo/ de adaptación para encontrar el mejor individuo de la población.
2. La selección de los padres se realiza por torneo de los individuos escogidos por pares aleatoriamente. Es decir, se escoge un par aleatorio de individuos y se realiza competición entre ellos (gana el mejor de la función adaptación).
3. Luego de la escogencia de los padres se somete la decisión de cruzamiento a probabilidades. La probabilidad de cruzamiento es un parámetro del algoritmo. Para ver si el cruzamiento es viable, se genera un número aleatorio entre 0 y 1 y si este número es menor o igual a la probabilidad de cruzamiento, se realiza el cruzamiento; de lo contrario, los hijos serán idénticos a los padres en la nueva población.

De realizarse el cruzamiento, se muestra a continuación la manera como se hace:

Operadores de cruzamiento

Para la parte binaria del encoding se realiza un cruzamiento básico, en el cual, se generó un número aleatorio entero entre $n+1$ y $3n$ (desde $n+1$ hasta $3n$ es la parte binaria del encoding), el cual se llamará punto de corte y dividirá la parte binaria del padre en 2 (la anterior y la posterior al punto de corte, las cuales llamaremos primera y segunda parte respectivamente).

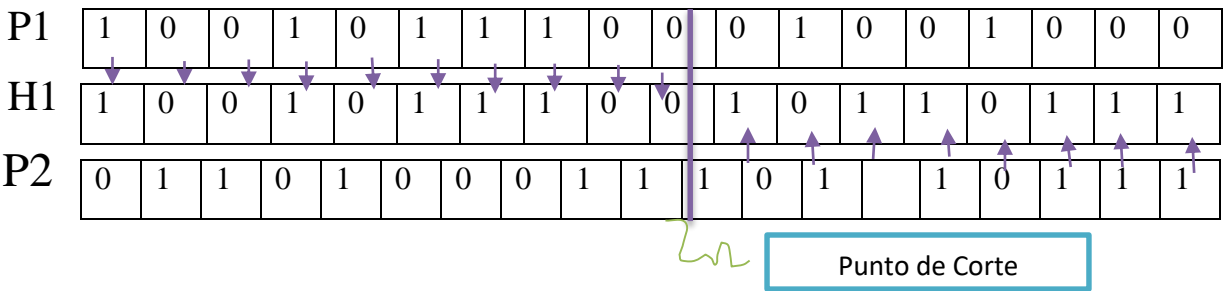
Una vez generado el punto de corte se realiza el traspaso del material genético de los padres a los hijos, esto se realiza de la siguiente forma: el Hijo (H1) estará

conformado por la primera parte del padre 1(P1) y la segunda parte del padre 2 (P2), para el hijo 2 (H2) se realiza el mismo proceso pero al contrario es decir la primera parte del H2 estará conformada por el padre 2 (H2) y la segunda parte estará conformada por el padre 1 (P1).

Este proceso se puede ver mejor en la Ilustración 7, en la cual se realiza el cruzamiento para un encoding de 9 instalaciones (se omite la parte permutada, pues será estudiada posteriormente). Supóngase el punto de corte como 10.

La parte permutada del encoding muestra la secuencia en la que se colocarán las piezas en la lámina. Para realizar el cruzamiento de la parte permutada no es suficiente con el cruzamiento básico, ya que podrían presentarse repeticiones en el encoding u omisión de instalaciones a la hora de la disposición, lo cual resulta en encoding ilegales para el sistema. El cruzamiento de este tipo de encoding no es materia nueva de estudio en algoritmos, ya que este tipo de representación de soluciones se ve en otro tipo de problemas como el Schedulling. Los cruces más destacados se listan en la tabla 1

Ilustración 7: Cruzamiento para la parte binaria del encoding



P1



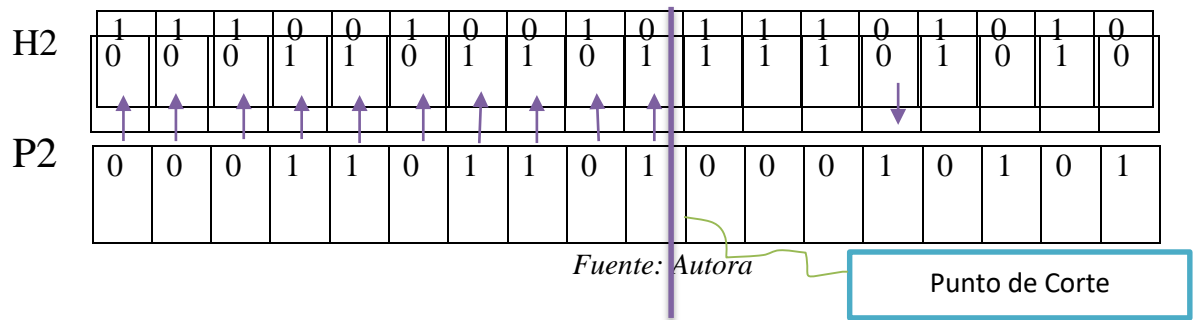


Tabla 1: Principales operadores de cruzamiento de encodings permutados

| ABREVIATURA | NOMBRE |
|-------------|---------------------------------------|
| OX | Order Crossover |
| PMX | Partial Mapping Crossover |
| UOB | Uniform Order Based |
| OP | One-point Order Crossover |
| GPX | Generalized Position Crossover |
| TP | Two-point Order Crossover |
| SJOX | Similar Job Order Crossover |
| SJ2OX | Similar Job 2-Point Order Crossover |
| SBOX | Similar Block Order Crossover |
| SB2OX | Similar Block 2-Point Order Crossover |

Fuente: (Ruiz, Maroto, & Alcaraz, 2006)

En este trabajo se utilizó el cruzamiento SJ2OX el cual demostró ser eficiente y guarda mejor el principio de transmisión del material genético en trabajos por (Álvarez & Noble, 2013)

Cruzamiento SJ2OX

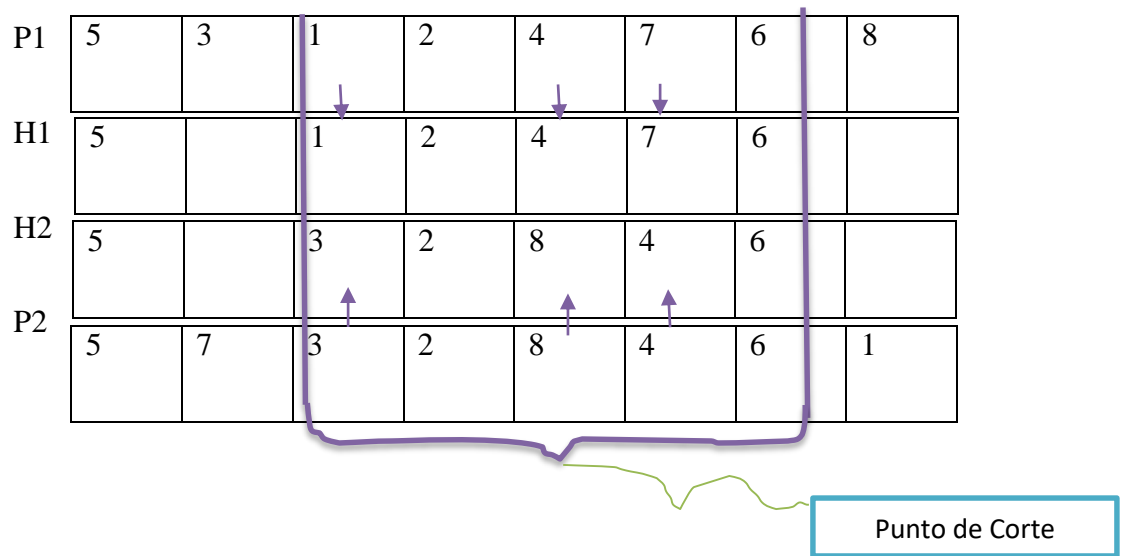
El SJ2OX fue propuesto por Ruiz et al. (2006) es muy similar a su preliminar SJOX sólo que este posee dos puntos de corte en vez de uno como el SJOX. Los padres le transmiten el material genético que es idéntico en los dos a sus hijos, luego de esto se seleccionan aleatoriamente dos puntos de corte. Cada uno de los padres le transfiere a su hijo los genes, entre los puntos de corte, es decir el padre 1 al hijo 1 y el padre 2 al hijo 2, y el resto se coloca en el orden en que aparezcan en el otro padre, ósea del padre 1 al hijo 2 y del padre 2 al hijo 1. Para ilustrar mejor el cruzamiento SJ2OX véase la Ilustración 7 para un encoding de 8 piezas (se omite la parte binaria).

1) Copia de los genes idénticos en los 2 padres

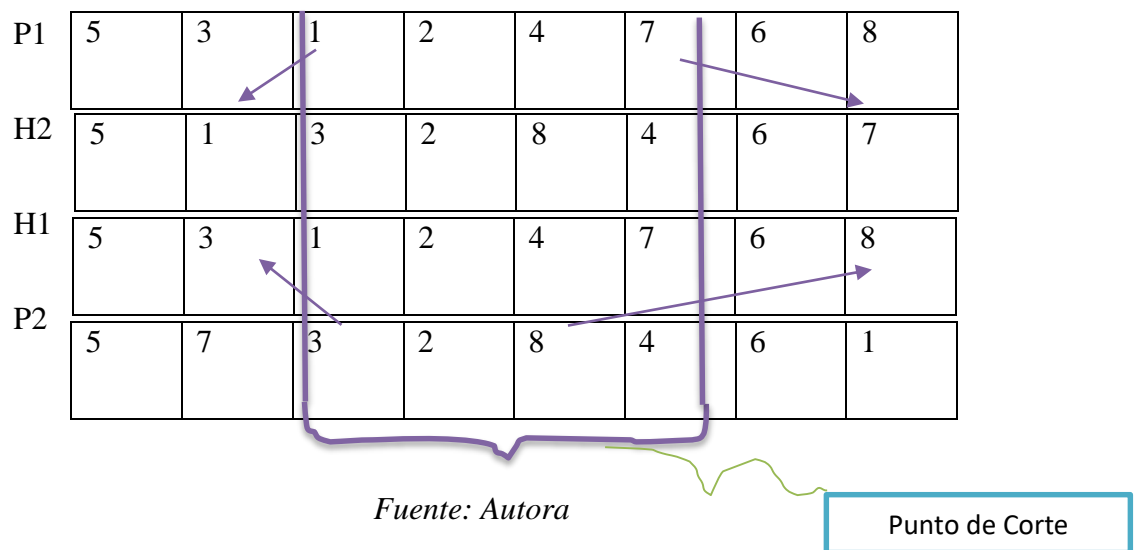
Ilustración 8: Cruzamiento SJ2OX para un encoding de 8 instalaciones

| | | | | | | | | |
|----|---|---|---|---|---|---|---|---|
| P1 | 5 | 3 | 1 | 2 | 4 | 7 | 6 | 8 |
| H1 | 5 | | | 2 | | | 6 | |
| H2 | 5 | | | 2 | | | 6 | |
| P2 | 5 | 7 | 3 | 2 | 8 | 4 | 6 | 1 |

2) Transferencia de los genes internos de padres hacia hijos



3) Relleno del nuevo hijo con los genes faltantes en el orden



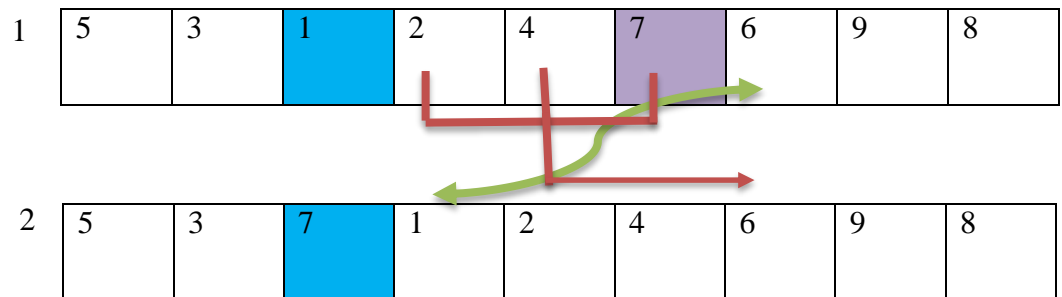
- Realizando un algoritmo similar al de la determinación de la viabilidad del cruzamiento de los padres se realiza un sondeo para determinar si se muta un gen de los hijos en la nueva población. La probabilidad de mutación también es un parámetro del GA. Para la mutación de los individuos también se tuvo que

discriminar la parte permutada de la binaria del encoding como se ve a continuación:

Operador de mutación

- I. En la parte binaria, la mutación se realiza de forma clásica. De ser viable, la forma de mutar un gen se hace mediante el cambio de 0 por 1 o al contrario es decir unos por ceros dependiendo del gen a mutar. Primero se determina el gen a mutar de manera aleatoria (un número aleatorio entre $n+1$ y $3n$) y luego si ese gen es un 0 se cambia por 1, si es el contrario se cambiaría entonces 1 por 0.
- II. En este trabajo para la mutación en la parte permutada se hará la forma Shift, la cual trata de seleccionar un gen al azar y lo cambia a una posición determinada aleatoriamente, los genes restantes se corren una posición a la derecha o izquierda de su posición original si el gen mutado se mueve a la izquierda o a la derecha respectivamente. Este se puede entender mucho mejor en la Ilustración 9 con un encoding para 9 piezas. (se omite la parte binaria).

Ilustración 9: Operador Shift para un encoding de 9 piezas



Fuente: Autora

5. Se debe repetir los pasos anteriores (3 y 4) hasta completar la nueva población y evaluar nuevamente la función de adaptación de los hijos a ver si se mejoró o no su adaptación de los hijos a ver si se mejoró o no su adaptación. Una vez hecho esto se escogen nuevamente los padres de una nueva generación. Para repetir el proceso si no se cumplió el criterio de parada.

Definición de parámetros del algoritmo genético

Para un adecuado funcionamiento del algoritmo se han tomado los siguientes valores para los parámetros, estos valores se han tomado debido a que la amplia utilización de esta metaheurística ha generalizado estos valores de los parámetros.

Tabla 2: Valores usados de los parámetros del Algoritmo Genético

| Parámetro | Valor |
|-----------------------------|--------------|
| Probabilidad de cruzamiento | 0,95 |
| Probabilidad de mutación | 0,05 |

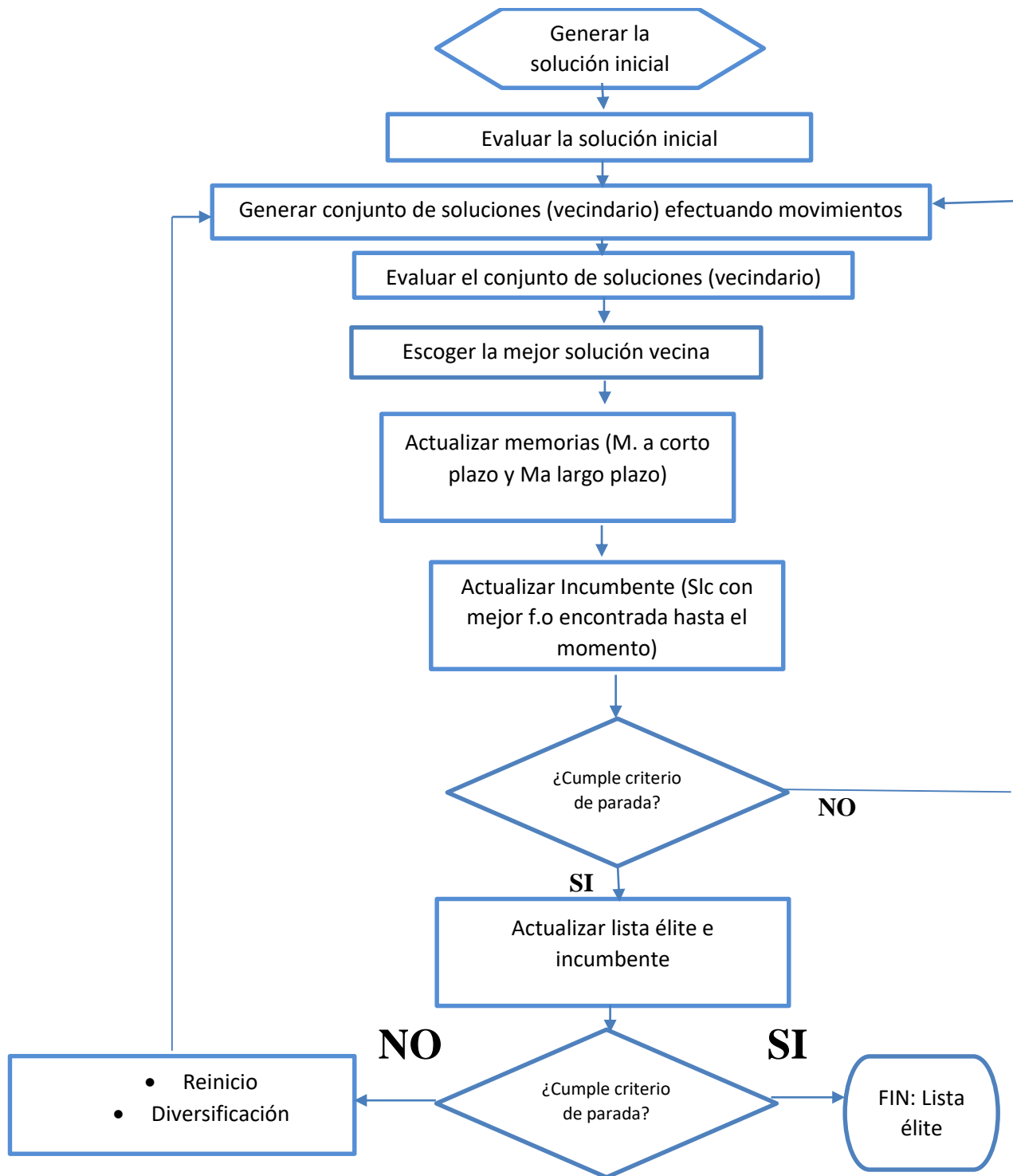
Fuente: Pendiente

3.2.2. Búsqueda Tabú (Tabu Search-TS)

El término Búsqueda Tabú (Tabu Search-TS) fue introducido en 1996 por Fred Glover. La búsqueda Tabú es una metaheurística que guía un procedimiento heurístico de búsqueda local en la búsqueda de optimalidad global. Su filosofía se basa en derivar y explotar una colección de estrategias inteligentes para la resolución de problemas, basada en procedimientos implícitos y explícitos de aprendizaje.

La búsqueda se enmarca dentro de las estrategias que afectan soluciones que empeoran el valor de la solución actual, básicamente la búsqueda tabú se basa en el uso de memoria y ese es el componente básico de la búsqueda tabú, la memoria se usa para no repetir trayectorias dentro del espacio de búsqueda y existen dos tipos de memoria; memoria a corto plazo y memoria a largo plazo, la memoria a corto plazo se utiliza para hacer intensificación en la búsqueda, es decir vamos a tratar de buscar buenas soluciones cercanas a la solución actual, la memoria a largo plazo por el contrario se utiliza para ir a lugares del espacio de búsqueda que no han sido visitados por el momento. En la memoria se guardan solo atributos de las soluciones. La Búsqueda Tabú es un conjunto de ideas que se juntan para resolver un problema de optimización. Para entender mejor el algoritmo de búsqueda Tabú a continuación se mostrará cómo opera: Dada una solución inicial, se examina un conjunto de soluciones vecinas de la solución inicial y se selecciona la mejor de entre las vecinas, que pasa a ser la nueva solución (aun cuando no sea mejor que la solución de partida). Sin embargo, a la medida que valla pasando el tiempo van haber unas soluciones que están prohibidas por lo tanto no pueden ser soluciones candidatas al menos que si esta solución prohibida mejora la función objetivo se hace criterio de aspiración y puede ser seleccionada. Este proceso se repite iterativamente hasta que se cumpla un criterio de parada. Para entender mejor este algoritmo se muestra el siguiente diagrama de flujo.

Ilustración 10: Diagrama del Algoritmo de Búsqueda Tabú

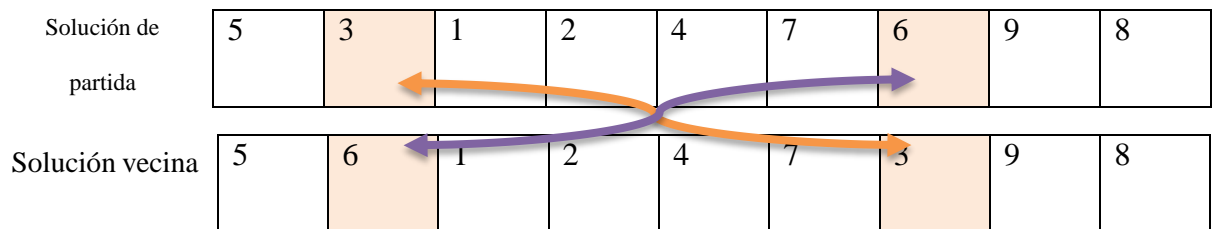


Fuente: Autora

Luego de este diagrama de flujo se procederá a listar en la siguiente secuencia de pasos los detalles de las etapas del proceso

1. Se genera una solución inicial y se evalúa la función objetivo/ de adaptación.
2. Luego se genera un conjunto de soluciones (vecinos) efectuando movimientos, estos movimientos en este trabajo son realizados por medio del operador de mutación; en la parte binaria, la mutación se realiza de forma clásica. De ser viable, la forma de mutar un gen se hace mediante el cambio de 0 por 1 o al contrario es decir unos por ceros dependiendo del gen a mutar. Primero se determina el gen a mutar de manera aleatoria (un número aleatorio entre $n+1$ y $3n$) y luego si ese gen es un 0 se cambia por 1, si es el contrario se cambiaría entonces 1 por 0. Y en la permutada se hace uso del Swap el cual consiste en intercambiar dos números de sus posiciones originales, los cuales son escogidos aleatoriamente. Swap se puede entender mejor observando la ilustración 11 en un encoding realizado para 9 piezas.

Ilustración 11: Operador Swap para un encoding de 9 piezas



Fuente: Autora

3. A este conjunto de vecinos se les hace su respectiva evaluación. De aquí se hace la escogencia de la mejor solución vecina, viendo si esta mejora la solución

encontrada por el momento, si la mejora, es escogida y enseguida se mira si se encuentra en Lista tabú, si está en lista tabú y mejora la función objetivo se hace el uso del criterio de aspiración y se actualiza la solución si la nueva solución tomada es mejor que cualquiera de las candidatas por lo tanto se ejecuta el movimiento de la solución actual a la mejor solución candidata registrada. Si por el contrario la mejor solución vecina escogida no se encuentra en lista tabú enseguida se actualiza la solución si la nueva solución tomada es mejor que cualquiera de las candidatas y se ejecuta el movimiento de la solución actual a la mejor solución candidata registrada.

Al contrario de lo anterior es decir si ninguna de las soluciones vecinas es mejor que la mejor solución por el momento se procederá a calcular el valor de movimiento el cual está definido de la siguiente manera:

$$Vm = Fvecinos - Factual \quad \text{Ecuación 1}$$

Luego de esto se procede a calcular el valor de la frecuencia, la cual es el número de veces que se ha hecho ese movimiento. Y por último se calcula el valor de penalización el cual está definido de la siguiente manera:

$$VP = Vmovimiento + frecuencia \quad \text{Ecuación 2}$$

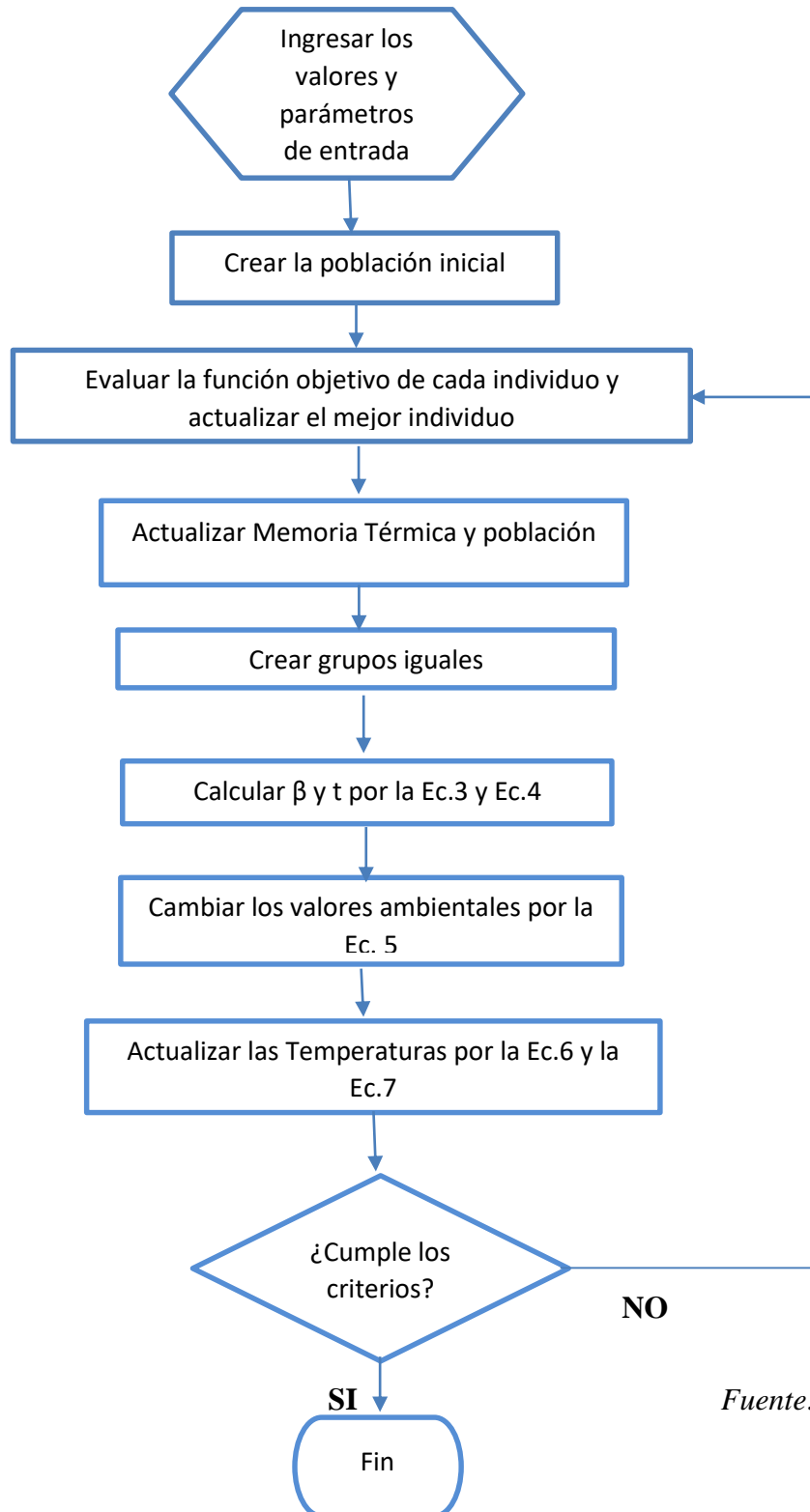
4. Sí cumple criterio de parada se finaliza el algoritmo, si sucede lo contrario se vuelve a generar un conjunto de soluciones vecinas y vuelve al ciclo nuevamente.

3.2.3. Optimización del Intercambio Térmico (TEO)

El algoritmo TEO está basado en los principios de la física. En este, cada agente se considera como un objeto de enfriamiento y al asociar otro agente como fluido

circundante, Una transferencia de calor y un intercambio térmico sucede entre ellos. Este algoritmo utiliza la ley de Newton de enfriamiento para actualizar las temperaturas. El proceso se repite hasta la satisfacción de las condiciones de terminación. Para más detalles se ilustra en el siguiente diagrama de flujo

Ilustración 12: Diagrama de Optimización del intercambio térmico



Fuente: Autora

Para el problema de este trabajo y dadas las condiciones del encoding a continuación se dará explicación de cada paso del diagrama de flujo expuesto anteriormente.

Paso 1. Se ingresan todas las variables de entradas del problema y los parámetros utilizados con sus respectivos valores, como lo son el Ancho y Largo de la lámina y las dimensiones de las piezas demandas y los parámetros que se definirán más adelante.

Paso 2. La generación de la población inicial se realizó de manera aleatoria.

Paso 3. Luego se hace la respectiva evaluación donde la función objetivo calcula el valor del desperdicio de cada parámetro de corte.

Paso 4. Teniendo en cuenta una memoria que guarde algunos mejores vectores históricos y sus correspondientes funciones objetivos, pueden mejorar el algoritmo sin aumentar el costo computacional. Para ello se utiliza una memoria térmica (MT), la cual guarda una serie de las mejores soluciones hasta ahora. Por lo tanto, en esta etapa, los vectores de solución guardados en MT se añaden a la población y se eliminan los mismos números de individuos peores actuales. Finalmente, los individuos se ordenan de acuerdo con sus correspondientes valores de f.o en un orden de menor a mayor.

Paso 5. Creación de grupos: Aquí los agentes se dividen en dos grupos iguales.

Paso 6. Definición de β . En la naturaleza cuando un individuo tiene β inferior, intercambia ligeramente la T° . Al inspirar esta característica se sugiere una fórmula similar. El valor de β para cada individuo se evalúa de acuerdo con la Ecuación 3 En esto, el individuo de menor costo tiene β menor y cambia ligeramente la posición.

$$\beta = \frac{\text{función objetivo del individuo}}{\text{individuo con peor función objetivo}} \quad \text{Ecuación 3}$$

Paso 7. Definición. El tiempo se asocia con el número de iteraciones. En esta formulación el valor de t para cada individuo se evalúa de acuerdo con la Ecuación 4 como:

$$t = \frac{toc}{tiempo\ máximo} \quad \text{Ecuación 4}$$

Paso 8. Para escapar de las trampas, es decir cuando los individuos se acercan a un óptimo local la temperatura ambiental es cambiada por la Ecuación 5

$$TAC = (1 - c1 + c2 * (1 - t) * rand) * TAC' \quad \text{Ecuación 5}$$

Donde $c1$ y $c2$ son las variables de control. TAC' Es la temperatura anterior del individuo que se modifica a TAC

- $(1-t)$ se considera que disminuye la trayectoria acercándose a las ultimas iteraciones. Al acercarse al final del proceso, t aumenta, lo que conduce a una disminución lineal en la aleatoriedad y el aumento de la explotación
- $C2(1-t)$. Esto puede considerarse igual a 0, cuando la disminución no es necesaria
- $C1$ controla el tamaño de los pasos aleatorios, además, cuando no se emplea un proceso decreciente ($c2=0$, como se ha dicho anteriormente) $c1$ implica la aleatoriedad

Aquí, $c=0(c1=c2=0)$, ningunos de los mecanismos mencionados anteriormente se emplean y 1 se multiplica con la temperatura anterior. En este trabajo $c1$ y $c2$ se eligen entre $\{0 \text{ y } 1\}$.

Paso 9. Actualización de los individuos. De acuerdo con los pasos anteriores y la Ecuación 5, la nueva temperatura de cada individuo es actualizada por la siguiente Ecuación 6.

$$T_{new} = TAC + (Told - TAC)\exp(-\beta t) \quad \text{Ecuación 6}$$

Donde T_{new} es un vector columna que nos indica cuantas veces se le debe hacer Swap a cada fila de la matriz MT

Paso 10. Segunda parte para escapar de óptimos locales: Aquí se introduce un parámetro Pro y se especifica si un componente de cada individuo de enfriamiento debe ser cambiado o no. Para cada individuo se compara con un aleatorio uniformemente distribuido dentro de (0,1) llamado Ran. Si Ran es menor que Pro se hace mutación Shift, luego de esto se escoge un aleatorio de la matriz térmica (TM) si este aleatorio cae en la parte binaria se hace la mutación clásica es decir si cae en un 0 este se cambia por 1 ó lo contrario, pero si por el contrario es decir si este aleatorio cae en la parte permutada se mira cual es el min y el Max de esa columna, estos valores son colocados en la Ecuación 6, Con el valor que dé en esta Ecuación. Se mete en la Ecuación 7. de mod, la cual da el otro valor con el que se va a mutar, por último, se hace shift con los dos valores.

$$T_{i,j} = T_{j,min} + rand(T_{j,max} - T_{j,min}) \quad \text{Ecuación 7}$$

Donde $T_{i,j}$ es la j-esima variable de i-ésimo individuo. $T_{j,min}$ y $T_{j,max}$ respectivamente, son los limites inferior y superior de la j-esima variable.

Paso 11. Comprobación de las condiciones de terminación. El proceso de optimización terminará después de un tiempo definido. Si el criterio no se cumple, pasa al Paso 3 para una nueva ronda de tiempo, de lo contrario el proceso se para e informara de la mejor solución encontrada.

Definición de parámetros del algoritmo TEO

Para un adecuado funcionamiento del algoritmo se han tomado los siguientes valores para los parámetros, estos valores se han tomado debido a que la amplia utilización de esta metaheurística ha generalizado estos valores de los parámetros.

Tabla 3: Valores usados de los parámetros del Algoritmo TEO

| Parámetro | Valor |
|------------------|--------------|
| Pro | 1 |

Fuente: Pendiente

CAPITULO 4: ANÁLISIS DE RESULTADOS Y CONCLUSIONES

4.1. ANÁLISIS DE RESULTADOS

Para realizar el análisis de los resultados se toman instancias del problema de corte bidimensional tipo guillotina encontradas en la literatura y se comparan los resultados. Sin embargo, dado que en la literatura no se encuentran problemas que aborden la optimización del problema de corte bidimensional tipo guillotina con el uso de varias láminas de igual dimensión se toma para esta parte los resultados obtenidos de las corridas, y se realiza una comparación de los algoritmos probados.

4.1.1. ANÁLISIS DE COMPARACIÓN GLOBAL DE ALGORITMOS

Los resultados arrojados con la implementación de los algoritmos genético, Búsqueda Tabú y Optimización del intercambio térmico (TEO) son mostrados en la tabla 4, de la cual se analiza que:

Tabla 4: Comparación de Algoritmos por Instancias proporcionadas por la literatura

| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BÚSQUEDA TABÚ | | | ALGORITMO TEO | | |
|------------|--------------------|-----------------|-----------|----------------------------|-----------------|-----------|---------------|-----------------|-----------|
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| ATP10 | 2691674,4 | 927491,982 | 2452197 | 2452197 | 0 | 2452197 | 2452197 | 0 | 2452197 |
| ATP11 | 5301526 | 0 | 5301526 | 5301526 | 0 | 5301526 | 5301526 | 0 | 5301526 |
| ATP12 | 5332765 | 0 | 5332765 | 5332765 | 0 | 5332765 | 5332765 | 0 | 5332765 |
| ATP13 | 5046655 | 1231058,713 | 2014483 | 2014483 | 0 | 2014483 | 2014483 | 0 | 2014483 |
| ATP14 | 4810110,667 | 2307482,166 | 2724848 | 2724848 | 0 | 2724848 | 2724848 | 0 | 2724848 |
| ATP15 | 8326544 | 0 | 8326544 | 7922732,533 | 1563955,085 | 2269372 | 8326544 | 0 | 8326544 |
| ATP16 | 8568929 | 0,094491118 | 8568929 | 8568929 | 0,094491118 | 8568929 | 8568929 | 0,094491118 | 8568929 |
| ATP17 | 4166571 | 0 | 4166571 | 4166571 | 0 | 4166571 | 4166571 | 0 | 4166571 |
| ATP18 | 4219204 | 2673817,605 | 2661504 | 2661504 | 0 | 2661504 | 2661504 | 0 | 2661504 |
| ATP19 | 7223288 | 0 | 7223288 | 7223288 | 0 | 7223288 | 7223288 | 0 | 7223288 |

| | | | | | | | | | |
|------------|-------------|-------------|----------|----------|-------------|----------|----------|---|----------|
| B | 12084492 | 2323790,008 | 3684492 | 3684492 | 0 | 3684492 | 3684492 | 0 | 3684492 |
| H | 8663 | 0 | 8663 | 8663 | 0 | 8663 | 8663 | 0 | 8663 |
| HZ1 | 2237 | 0 | 2237 | 2237 | 0 | 2237 | 2237 | 0 | 2237 |
| M1 | 10008 | 0 | 10008 | 10008 | 0 | 10008 | 10008 | 0 | 10008 |
| M2 | 62297 | 0 | 62297 | 62297 | 0 | 62297 | 62297 | 0 | 62297 |
| M3 | 95703 | 0 | 95703 | 95703 | 0 | 95703 | 95703 | 0 | 95703 |
| M4 | 179828 | 0 | 179828 | 179828 | 0 | 179828 | 179828 | 0 | 179828 |
| M5 | 387216 | 0 | 387216 | 387216 | 0 | 387216 | 387216 | 0 | 387216 |
| U1 | 17496046 | 0 | 17496046 | 17496046 | 0 | 17496046 | 17496046 | 0 | 17496046 |
| U2 | 10178804 | 0 | 10178804 | 10178804 | 0 | 10178804 | 10178804 | 0 | 10178804 |
| U4 | 38417305 | 0 | 38417305 | 38417305 | 0 | 38417305 | 38417305 | 0 | 38417305 |
| UU1 | 573478,3333 | 121987,5091 | 490145 | 340145 | 126773,1382 | 240145 | 490145 | 0 | 490145 |
| UU2 | 1068757 | 211119,4652 | 988757 | 948757 | 154919,3338 | 388757 | 988757 | 0 | 988757 |
| UU3 | 1774681 | 0 | 1774681 | 894681 | 455443,2692 | 674681 | 1774681 | 0 | 1774681 |
| UU4 | 2892985 | 309838,6677 | 1772985 | 1772985 | 0 | 1772985 | 1772985 | 0 | 1772985 |

| | | | | | | | | | |
|----------------|--------------------|--------------------|-------------|--------------------|--------------------|-------------|--------------------|--------------------|-------------|
| UU5 | 5630022 | 0 | 5630022 | 3745022 | 0 | 3745022 | 3745022 | 0 | 3745022 |
| UU6 | 8700030 | 0 | 8700030 | 5713180 | 0 | 5713180 | 5713180 | 0 | 5713180 |
| UU7 | 9166222,667 | 1531202,04 | 7584804 | 5212676 | 1227692,876 | 4619644 | 7584804 | 0 | 7584804 |
| UU8 | 12242789,33 | 1407463,101 | 11709456 | 7709456 | 0 | 7709456 | 11442789,33 | 1032795,559 | 7709456 |
| UU9 | 25743318 | 0 | 25743318 | 15493318 | 3000892,724 | 13443318 | 19593318 | 0 | 19593318 |
| UU10 | 35091479 | 4248779,236 | 33481479 | 23016479 | 4248779,236 | 21406479 | 33481479 | 0 | 33481479 |
| UU11 | 20216407 | 0 | 20216407 | 7038907 | 0,094491118 | 7038907 | 7038907 | 0,094491118 | 7038907 |
| Total | 8053438,606 | 9685976,369 | 2237 | 5961782,767 | 7984105,288 | 2237 | 6653791,292 | 8978326,124 | 2237 |
| general | | | | | | | | | |

Fuente: Autora

- ✓ El algoritmo de Búsqueda Tabú fue el ganador frente a los otros dos algoritmos (TEO y Genético) en las instancias ATP15, UU1, UU2, UU3, UU7, UU8, UU9, UU10 lo que representa un 25% del total de instancias
- ✓ El algoritmo de Búsqueda Tabú quedo empate frente al algoritmo TEO en las instancias ATP10, ATP13, ATP14, ATP18, B, UU4, UU5, UU6 y UU11 lo que representa un 28.1% del total de instancias

- ✓ El algoritmo genético solo quedo empate con los otros dos algoritmos (Búsqueda Tabú y TEO) en las instancias ATP11, ATP12, ATP16, ATP17, ATP19, H, HZ1, M1, M2, M3, M4, M5, U1, U2 y U4 lo que representa un 46.8% del total de instancias.

4.2.ANÁLISIS DE COMPARACIÓN POR NÚMERO DE PIEZAS EN LAS INSTANCIAS

Ahora realizamos una comparación en la que se hace uso del número de piezas que contiene cada instancia, es decir que se dividen las instancias en grupos según el número de piezas que contengan las instancias. Lo que se quiere notar en esta parte es ver que tanto afecta el número de piezas al ser corridos los algoritmos. A continuación, se muestran los grupos y el análisis según los resultados mostrados en las tablas 5, 6, 7, 8, 9 y 10.

| Tabla 5: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS H y HZ1 | | | | | | | | | |
|---|--------------------|-----------------|-----------|----------------------------|-----------------|-----------|---------------|-----------------|-----------|
| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BUSQUEDA TABÚ | | | ALGORITMO TEO | | |
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| H | 8663 | 0 | 8663 | 8663 | 0 | 8663 | 8663 | 0 | 8663 |
| HZ1 | 2237 | 0 | 2237 | 2237 | 0 | 2237 | 2237 | 0 | 2237 |
| Total general | 5450 | 3267,927056 | 2237 | 5450 | 3267,927056 | 2237 | 5450 | 3267,927056 | 2237 |

Fuente: Autora

- ✓ Primero son comparados los tres algoritmos en las instancias H y HZ1 las cuales tienen un número de piezas de 5 y 6 respectivamente, al comparar los resultados mostrados en la tabla 5 se puede notar que en esta clase de instancias que contienen este número de piezas no se tiene variedad de resultado al ser corridos los algoritmos ya que el valor de la función objetivo en todas las corridas de cada algoritmo es el mismo.

| Tabla 6: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS M1, M2, M3, M4, M5, U1, U2 | | | | | | | | | |
|--|--------------------|--------------------|--------------|----------------------------|--------------------|--------------|--------------------|--------------------|--------------|
| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BUSQUEDA TABÚ | | | ALGORITMO TEO | | |
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| M1 | 10008 | 0 | 10008 | 10008 | 0 | 10008 | 10008 | 0 | 10008 |
| M2 | 62297 | 0 | 62297 | 62297 | 0 | 62297 | 62297 | 0 | 62297 |
| M3 | 95703 | 0 | 95703 | 95703 | 0 | 95703 | 95703 | 0 | 95703 |
| M4 | 179828 | 0 | 179828 | 179828 | 0 | 179828 | 179828 | 0 | 179828 |
| M5 | 387216 | 0 | 387216 | 387216 | 0 | 387216 | 387216 | 0 | 387216 |
| U1 | 17496046 | 0 | 17496046 | 17496046 | 0 | 17496046 | 17496046 | 0 | 17496046 |
| U2 | 10178804 | 0 | 10178804 | 10178804 | 0 | 10178804 | 10178804 | 0 | 10178804 |
| Total, general | 4058557,429 | 6518596,241 | 10008 | 4058557,429 | 6518596,241 | 10008 | 4058557,429 | 6518596,241 | 10008 |

Fuente: Autora

- ✓ El segundo grupo está constituido por las instancias M1, M2, M3, M4, M5, U1 y U2 éstas constan de 10 piezas cada una, se puede observar en la tabla 6 que en los tres algoritmos el valor de la función objetivo en todas las corridas de cada instancia es el mismo, lo que quiere decir que en el tipo de instancias que contienen este número de piezas el resultado es el mismo en los tres algoritmos.

Tabla 7: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS UU1, UU3, UU11

| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BUSQUEDA TABÚ | | | ALGORITMO TEO | | |
|---------------|--------------------|--------------------|---------------|----------------------------|--------------------|---------------|--------------------|--------------------|---------------|
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| UU1 | 573478,3333 | 121987,5091 | 490145 | 340145 | 126773,1382 | 240145 | 490145 | 0 | 490145 |
| UU3 | 1774681 | 0 | 1774681 | 894681 | 455443,2692 | 674681 | 1774681 | 0 | 1774681 |
| UU11 | 20216407 | 0 | 20216407 | 7038907 | 0,094491118 | 7038907 | 7038907 | 0,094491118 | 7038907 |
| Total general | 7521522,111 | 9091869,863 | 490145 | 2757911 | 3081437,011 | 240145 | 3101244,333 | 2865317,527 | 490145 |

Fuente: Autora

- ✓ El tercer grupo está constituido por las instancias UU1, UU3 y UU11, las cuales constan con un total de 25 piezas cada una, en este grupo según la tabla 7 se puede dar cuenta que en las dos primeras instancias (UU1 Y UU3) que representan un 66.7% del total de este grupo de instancias el algoritmo que minimiza la función objetivo es el de Búsqueda Tabú para cada una de las

instancias, en la tercera lo que representa un 33.3% del total de este grupo de instancias instancia el algoritmo Búsqueda Tabú y el TEO muestran mejor resultado es decir una mejor función objetivo.

Tabla 8: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP12, ATP18, ATP19, B, UU2, UU4, UU6

| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BÚSQUEDA TABÚ | | | ALGORITMO TEO | | |
|----------------------|--------------------|--------------------|---------------|----------------------------|--------------------|---------------|--------------------|--------------------|---------------|
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| ATP12 | 5332765 | 0 | 5332765 | 5332765 | 0 | 5332765 | 5332765 | 0 | 5332765 |
| ATP18 | 4219204 | 2673817,605 | 2661504 | 2661504 | 0 | 2661504 | 2661504 | 0 | 2661504 |
| ATP19 | 7223288 | 0 | 7223288 | 7223288 | 0 | 7223288 | 7223288 | 0 | 7223288 |
| B | 12084492 | 2323790,008 | 3684492 | 3684492 | 0 | 3684492 | 3684492 | 0 | 3684492 |
| UU2 | 1068757 | 211119,4652 | 988757 | 948757 | 154919,3338 | 388757 | 988757 | 0 | 988757 |
| UU4 | 2892985 | 309838,6677 | 1772985 | 1772985 | 0 | 1772985 | 1772985 | 0 | 1772985 |
| UU6 | 8700030 | 0 | 8700030 | 5713180 | 0 | 5713180 | 5713180 | 0 | 5713180 |
| Total general | 5931645,857 | 3705621,048 | 988757 | 3905281,571 | 2122041,919 | 388757 | 3910995,857 | 2113271,207 | 988757 |

Fuente: Autora

- ✓ El cuarto grupo consta de las instancias ATP12, ATP18, ATP19, B, UU2, UU4 y UU6, estas tienen un total de 35, 31, 35, 32, 30, 38, 38 de piezas respectivamente, en la tabla 8 se puede dar cuenta que las instancias ATP12, ATP19, UU6 que representa

un 42.8% del total de este grupo de instancias el resultado es totalmente igual en todos los algoritmos ósea que en estas instancias el valor de la función objetivo para todas las corridas en cada instancia es el mismo en los tres algoritmos. En las instancias ATP18, B, UU4 que representa un 42.8% del total de este grupo de instancias el valor de la función objetivo es igual en los tres algoritmos, pero en el algoritmo genético no en todas las corridas de cada una de las instancias el valor de la función objetivo no fue el mismo. Y por último en la instancia UU2 que representa un 14.3% del total de este grupo de instancias el algoritmo que muestra mejor resultado con respecto a la función objetivo es el Búsqueda Tabú

Tabla 9: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP14, ATP15, U4

| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BÚSQUEDA TABÚ | | | ALGORITMO TEO | | |
|----------------------|--------------------|-------------------|----------------|----------------------------|--------------------|----------------|--------------------|--------------------|----------------|
| INSTANCIA | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| ATP 14 | 4810110,667 | 2307482,166 | 2724848 | 2724848 | 0 | 2724848 | 2724848 | 0 | 2724848 |
| ATP15 | 8326544 | 0 | 8326544 | 7922732,533 | 1563955,085 | 2269372 | 8326544 | 0 | 8326544 |
| U4 | 38417305 | 0 | 38417305 | 38417305 | 0 | 38417305 | 38417305 | 0 | 38417305 |
| Total general | 17184653,22 | 15308090,7 | 2724848 | 16354961,84 | 15946419,93 | 2269372 | 16489565,67 | 15850094,57 | 2724848 |

Fuente: Autora

- ✓ El quinto grupo está conformado por las instancias ATP14, ATP15 y U4, las cuales tienen un total de 42, 49 y 40 piezas respectivamente, de acuerdo a los resultados arrojados en la tabla 9 se puede notar que en la instancia ATP14 que representa un

33.3% del total de este grupo de instancias, el resultado de la función objetivo es igual en los tres algoritmos, pero con la diferencia de que el valor de la función objetivo no fue el mismo en todas las corridas en el algoritmo genético. En la instancia ATP15 que representa un 33.3% del total de este grupo de instancias, el algoritmo que arroja un mejor resultado es el algoritmo de Búsqueda Tabú ya que este nos muestra un valor de la función objetivo más mínimo que el de los otros dos algoritmos. En la instancia U4 que representa un 33.3% del total de este grupo de instancias, en todas las corridas el valor de la función objetivo es el mismo en los tres algoritmos.

Tabla 10: COMPARACIÓN POR GRUPOS SEGÚN EL NÚMERO DE PIEZAS DE LAS INSTANCIAS ATP10, ATP 11, ATP13, ATP16, ATP17, UU5, UU7, UU8, UU9, UU10

| ALGORITMOS | ALGORITMO GENETICO | | | ALGORITMO DE BÚSQUEDA TABÚ | | | ALGORITMO TEO | | |
|--------------|--------------------|-----------------|-----------|----------------------------|-----------------|-----------|---------------|-----------------|-----------|
| INSTANCIAS | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f | Promedio de f | Desv. Est. de f | Mín. de f |
| ATP10 | 2691674,4 | 927491,98 | 2452197 | 2452197 | 0 | 2452197 | 2452197 | 0 | 2452197 |
| ATP11 | 5301526 | 0 | 5301526 | 5301526 | 0 | 5301526 | 5301526 | 0 | 5301526 |
| ATP13 | 5046655 | 1231058,7 | 2014483 | 2014483 | 0 | 2014483 | 2014483 | 0 | 2014483 |
| ATP16 | 8568929 | 0,0944911 | 8568929 | 8568929 | 0,094491 | 8568929 | 8568929 | 0,094491 | 8568929 |
| ATP17 | 4166571 | 0 | 4166571 | 4166571 | 0 | 4166571 | 4166571 | 0 | 4166571 |
| UU5 | 5630022 | 0 | 5630022 | 3745022 | 0 | 3745022 | 3745022 | 0 | 3745022 |

| | | | | | | | | | |
|----------------------|--------------------|-----------------|----------------|------------------|----------------|----------------|------------------|----------------|----------------|
| UU7 | 9166222,667 | 1531202 | 7584804 | 5212676 | 1227693 | 4619644 | 7584804 | 0 | 7584804 |
| UU8 | 12242789,33 | 1407463,1 | 11709456 | 7709456 | 0 | 7709456 | 11442789 | 1032796 | 7709456 |
| UU9 | 25743318 | 0 | 25743318 | 15493318 | 3000893 | 13443318 | 19593318 | 0 | 19593318 |
| UU10 | 35091479 | 4248779,2 | 33481479 | 23016479 | 4248779 | 21406479 | 33481479 | 0 | 33481479 |
| Total general | 11364918,64 | 10246112 | 2014483 | 7768065,7 | 6522783 | 2014483 | 9835111,8 | 9358977 | 2014483 |

Fuente: Autora

- ✓ El sexto grupo está compuesto por las instancias ATP10, ATP11, ATP13, ATP16, ATP17, UU5, UU7, UU8, UU9 y UU101, las cuales están conformadas por un total de 51, 58, 54, 53, 59, 50, 50, 55, 60 y 55 piezas respectivamente, en este grupo se puede notar en la tabla 10 que las instancias ATP10 y ATP13 que representa un 20% del total de este grupo de instancias, para cada instancia el resultado de la función objetivo es igual en los tres algoritmos, pero con la diferencia de que el valor de la función objetivo no fue el mismo en todas las corridas en el algoritmo genético. En las instancias ATP11, ATP16, ATP17 y UU5 que representa un 40% del total de este grupo de instancias, el resultado es totalmente igual en todos los tres algoritmos ósea que en estas instancias el valor de la función objetivo para todas las corridas en cada instancia es el mismo en los tres algoritmos. En las instancias UU7, UU9 y UU10 que representa un 30% del total de este grupo de instancias, el algoritmo de Búsqueda Tabú le gana a los otros dos algoritmos ya que los resultados muestran que el valor de la función fue mínima en estas clases de instancias. Y por último en la instancia UU8 que representa un 10% del total de este grupo de instancias, se puede notar en los resultados que el valor de la función objetivo es el mismo en los algoritmos de Búsqueda Tabú y TEO pero con la diferencia de que en el algoritmo de Búsqueda Tabú en las corridas de esta instancia fue el mismo en todas en cambio en el TEO no.

4.3.CONCLUSIONES Y TRABAJOS FUTUROS

En el desarrollo de esta investigación se realizó la adaptación de 3 algoritmos seleccionados en la metodología para la medición de sus desempeños y su posterior contraste.

La comparación de estos 3 algoritmos metaheurísticos aplicados al problema de corte bidimensional tipo guillotina se realizó con el fin de determinar cuál era el mejor en cuanto a calidad de soluciones. Dos de estos algoritmos han sido aplicados anteriormente al problema, mientras que con el tercero se incursionó por primera vez en la aplicación de esta metaheurística (Optimización del Intercambio Térmico-TEO-).

Luego de haber realizado el respectivo análisis de los resultados se puede afirmar que el algoritmo de Búsqueda Tabú ofrece buenas soluciones al problema de corte bidimensional tipo guillotina. Este algoritmo se caracterizó por ofrecer mejores soluciones que los otros dos algoritmos.

Por otro lado, se pudo observar que la metaheurística TEO cuya aplicación se realizó por primera vez para este tipo de problemas en esta investigación y en la cual se centra parte del interés de la misma ha dado resultados casi iguales a los arrojados por el algoritmo del Búsqueda tabú y perdió pocas veces en algunas instancias con él.

Con respecto al algoritmo genético se observaron resultados sobresalientes ya que en la mayoría de las instancias quedó empate frente a los otros dos algoritmos y en pocas instancias perdió.

Para trabajos futuros se propone, por un lado, la aplicación de los principios expuestos en este trabajo en ambientes reales mediante la colaboración interinstitucional para medir el

impacto de las mejoras propuestas por el algoritmo aplicado. En segunda instancia se propone la investigación de otros tipos de funciones objetivos.

Diseñar o adaptar algoritmos que solucionen otros tipos de problemas que son derivados del problema de corte y empaquetamiento (C&P) como los son el problema bidimensional de corte de stock de tamaño múltiple (MS2DSP).

CAPITULO 5: BIBLIOGRAFÍA

- Alvarez-Valdés, R., Parajón, A., & Tamarit, J. M. (2001). A Computational Study of Heuristic Algorithms for Two-Dimensional Cutting Stock Problems. *MIC 2001 - The 4th Metaheuristics International Conference*, 7–12. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.9465>
- Álvarez, A., & Noble, V. (2013). COMPARACIÓN DE METAHEURÍSTICAS APLICADAS AL PROBLEMA DE DISTRIBUCIÓN DE INSTALACIONES (FLP) CON ÁREAS DESIGUALES Y DIMENSIONES FIJAS.
- Amossen, R. R., & Pisinger, D. (2010). Multi-dimensional bin packing problems with guillotine constraints. *Computers and Operations Research*, 37(11), 1999–2006. <https://doi.org/10.1016/j.cor.2010.01.017>
- Aryanezhad, M.-B., Fakhim Hashemi, N., Makui, A., & Javanshir, H. (2012). A simple approach to the two-dimensional guillotine cutting stock problem. *Journal of Industrial Engineering International*, 8(1), 21. <https://doi.org/10.1186/2251-712X-8-21>
- Christofides, N., & Whitlock, C. (1977). An Algorithm for Two-Dimensional Cutting Problems. *Operations Research*, 25(1), 30–44. <https://doi.org/10.1287/opre.25.1.30>
- Cintra, G. F., Miyazawa, F. K., Wakabayashi, Y., & Xavier, E. C. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1), 59–83. <https://doi.org/10.1016/j.ejor.2007.08.007>
- Clautiaux, F., Jouglet, A., Carlier, J., & Moukrim, A. (2008). A new constraint

- programming approach for the orthogonal packing problem. *Computers and Operations Research*, 35(3), 944–959. <https://doi.org/10.1016/j.cor.2006.05.012>
- Cui, Y., & Zhao, Z. (2013). Heuristic for the rectangular two-dimensional single stock size cutting stock problem with two-staged patterns. *European Journal of Operational Research*, 231(2), 288–298. <https://doi.org/10.1016/j.ejor.2013.05.042>
- Delgadillo-Avila, R. S. (2007). Un Estudio Algorítmico Del Problema De Corte Y Empaquetado 2D.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2), 145–159. [https://doi.org/10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K)
- Faggioli, E., & Bentivoglio, C. A. (1998). Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, 110(97), 564–575. [https://doi.org/10.1016/S0377-2217\(97\)00268-3](https://doi.org/10.1016/S0377-2217(97)00268-3)
- Furini, F., & Malaguti, E. (2013). Models for the two-dimensional two-stage cutting stock problem with multiple stock size. *Computers and Operations Research*, 40(8), 1953–1962. <https://doi.org/10.1016/j.cor.2013.02.026>
- Furini, F., Malaguti, E., Medina Durán, R., Persiani, A., & Toth, P. (2012). A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size. *European Journal of Operational Research*, 218(1), 251–260. <https://doi.org/10.1016/j.ejor.2011.10.018>
- García Sánchez, Á. (2013). Técnicas metaheurísticas, 47. Retrieved from <http://www.iol.etsii.upm.es/arch/metaheurísticas.pdf>

- Gilmore, P. C., & Gomory, R. E. (1965). Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research*, 13(1), 94–120.
<https://doi.org/10.1287/opre.13.1.94>
- Gilmore P. C. and Gomory R. E. (1961). Linear Programming Approach Cutting Stock. *Operations Research*.
- Herz, J. C. (1972). Recursive Computational Procedure for Two-dimensional Stock Cutting. *IBM Journal of Research and Development*, 16(5), 462–469.
<https://doi.org/10.1147/rd.165.0462>
- Hifi, M., M'Hallah, R., & Saadi, T. (2009). Approximate and exact algorithms for the double-constrained two-dimensional guillotine cutting stock problem. *Computational Optimization and Applications*, 42(2), 303–326.
<https://doi.org/10.1007/s10589-007-9081-5>
- Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software*, 110, 69–84.
<https://doi.org/10.1016/j.advengsoft.2017.03.014>
- Kröger, B. (1995). Guillotineable bin packing: A genetic approach. *European Journal of Operational Research*, 84(3), 645–661. [https://doi.org/10.1016/0377-2217\(95\)00029-P](https://doi.org/10.1016/0377-2217(95)00029-P)
- Lu, H. C., & Huang, Y. H. (2015). An efficient genetic algorithm with a corner space algorithm for a cutting stock problem in the TFT-LCD industry. *European Journal of Operational Research*, 246(1), 51–65. <https://doi.org/10.1016/j.ejor.2015.04.044>
- Malaguti, E., Medina Durán, R., & Toth, P. (2014). Approaches to real world two-

- dimensional cutting problems. *Omega (United Kingdom)*, 47, 99–115.
<https://doi.org/10.1016/j.omega.2013.08.007>
- Melián, B., Pérez, J., & Vega, M. (2003). Metaheurísticas: una visión global.
Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 7, 19.
 Retrieved from www.aepia.org
- Michalewicz, Z., Janikow, C. Z., & Krawczyk, J. B. (1992). A modified genetic algorithm for optimal control problems. *Computers & Mathematics with Applications*, 23(12), 83–94. [https://doi.org/10.1016/0898-1221\(92\)90094-X](https://doi.org/10.1016/0898-1221(92)90094-X)
- Morabito, R., & Pureza, V. (2010). A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, 179(1), 297–315.
<https://doi.org/10.1007/s10479-008-0457-4>
- Onwubolu, G. C., & Mutingi, M. (2003). A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, 14(2), 209–218.
<https://doi.org/10.1023/A:1022955531018>
- Osman, I. H., Ct, K., & Kelly, J. P. (1996). Abstract :, 1–21.
- Peláez, J. I., & Red, D. L. La. (n.d.). Un Algoritmo Evolutivo Paralelo Adaptativo para el problema del Corte con Guillotina en 2D, (September 2015).
- Péter, A. (2017). A Genetic Solution for the Cutting Stock Problem, (48), 1–6.
- Picardie, U. De, Verne, J., & Sciences, U. F. R. (2010). A cooperative algorithm for constrained two-staged two-dimensional cutting problems Mhand Hifi * and Toufik

Saadi, 9(1), 104–124.

Piezas, C. D. E., Guillotinado, C. O. N., Loecelia, M., Ruvalcaba, G., Gabriel, J., & Medina, C. (2014). Heurística De Dos-Etapas Para El Problema De, 19–33.

Rodriguez, G. D., Mauricio, D. S., & Rodríguez, G. D. (2015). Un Algoritmo GRASP de corte guillotina 2D con Agrupamiento y Rotación Un Algoritmo GRASP de corte guillotina 2D con Agrupamiento y Rotación, (September).

Ruiz, R., Maroto, C., & Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, 34(5), 461–476.
<https://doi.org/10.1016/j.omega.2004.12.006>

Santis, D. De, Sthefany, B., Cuesta, V., Guillermo, I., & Palma, B. (n.d.). “ Diseño de una heurística para resolver el problema de corte bidimensional rectangular en una empresa metalúrgica ” Resumen, (1).

Silva, E., Alvelos, F., & Valério de Carvalho, J. M. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, 205(3), 699–708.
<https://doi.org/10.1016/j.ejor.2010.01.039>

Terashima-Marín, H., Farías-Zárate, C. J., Ross, P., & Valenzuela-Rendón, M. (2007). Comparing Two Models to Generate Hyper-heuristics for the 2D-Regular Bin-Packing Problem. *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'07)*, 2182–2189. <https://doi.org/10.1145/1276958.1277377>

Tiwari, S., & Chakraborti, N. (2006). Multi-objective optimization of a two-dimensional cutting problem using genetic algorithms. *Journal of Materials Processing*

Technology, 173(3), 384–393. <https://doi.org/10.1016/j.jmatprotec.2005.12.011>

Toro-ocampo, E. M. (2010). Problema de la mochila irrestricta bidimensional guillotizada 1, *14*(2), 327–344.

Toro, O., Eliana, M., Rueda, M., & Augusto, C. (2007). Tabu Search algorithm for solving the two dimensional guillotined problem.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109–1130. <https://doi.org/10.1016/j.ejor.2005.12.047>

Yanasse, H. H., & Morabito, R. (2008). A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*, 46(21), 6189–6206.
<https://doi.org/10.1080/00207540601011543>

Zanakis, S. H., & Evans, J. R. (1981). Heuristic “ Optimization ”: Why , When , and How to Use It WHY , WHEN , AND HOW TO USE IT. *Interfaces*, 11(5)(August 2015), 84–91.

Vancells Flotats, J. (2002). *Algoritmos y programas*. UOC/Digitalia.

